

北京市高等教育精品教材立项项目  
新编计算机类本科规划教材

# Visual FoxPro 程序设计实用教程

刘 丽 金晓龙 编著

電子工業出版社  
Publishing House of Electronics Industry  
北京 · BEIJING

## 内 容 简 介

本书按照项目驱动的方式,以知识点和实例为主线介绍了 Visual FoxPro 的各种基本操作,并将一个系统开发实例贯穿到每个章节中,最终完成该系统的开发。本教材共分为 10 章,分别介绍了数据库系统及 Visual FoxPro 的基本概念、项目管理器及其操作、数据表的基本操作、数据库的设计与操作、程序设计基础、面向对象程序设计、表单的建立与使用、结构化查询语言(SQL)、查询与视图、报表设计、菜单设计、应用程序的生成和发布等,并配有丰富的例题、习题。本书不仅给出了系统设计步骤和源代码,而且突出介绍了各部分的设计思路、实现目标和设计流程,力求使读者抓住本质,迅速掌握使用 Visual FoxPro 开发管理信息系统的相关知识与技术。

本书较好地学习与实践结合在一起,既可作为高等院校非计算机专业应用型本科学生的教材,也可作为希望尽快开发小型数据库应用程序的技术人员的参考书。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。  
版权所有,侵权必究。

### 图书在版编目(CIP)数据

Visual FoxPro 程序设计实用教程 / 刘丽, 金晓龙编著. —北京: 电子工业出版社, 2009.6

新编计算机类本科规划教材

ISBN 978-7-121-08686-1

I. V… II. ①刘…②金… III. 关系数据库—数据库管理系统, Visual FoxPro 6.0—程序设计—高等学校—教材 IV. TP311.138

中国版本图书馆 CIP 数据核字(2009)第 061057 号

策划编辑: 冯小贝

责任编辑: 冯小贝

印 刷:

装 订:

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1092 1/16 印张: 19.5 字数: 499 千字

印 次: 2009 年 6 月第 1 次印刷

定 价: 29.00 元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 [zlts@phei.com.cn](mailto:zlts@phei.com.cn), 盗版侵权举报请发邮件至 [dbqq@phei.com.cn](mailto:dbqq@phei.com.cn)。

服务热线: (010) 88258888。

# 前 言

在数据库技术飞速发展的今天，Visual FoxPro 作为优秀的新一代小型数据库管理系统，仍然受到众多用户的青睐。Visual FoxPro 具有操作界面友好、功能完善和强大、语言简练、简单易学、辅助开发工具丰富、便于实现应用系统快速开发等特点，深受许多小型数据库应用系统开发人员的喜爱。为了满足高等学校 Visual FoxPro 程序设计类课程的教学需要，我们编写了这本《Visual FoxPro 程序设计实用教程》。作为北京市精品教材立项项目，在项目实施的过程中，项目组成员为了能够编写出优质的教材做出了许多努力。在编写本书的过程中，依据我们多年的实际教学经验，并且参考和借鉴了多本相关的同类教材，将一个系统开发实例贯穿到每个章节中，并最终完成该系统的开发。通过对各个具体实例的精心解析，读者不仅能够全面了解 Visual FoxPro 的基本风格，学会基本操作技能，同时还能从其各具示范性的操作过程中获得有益的启示，积累自己的开发经验。本书不仅给出了系统设计步骤和源代码，而且突出介绍了各部分的设计思路、实现目标和设计流程，力求使读者抓住本质，迅速掌握使用 Visual FoxPro 开发管理信息系统的相关知识，学到设计管理信息系统的思路和方法。本书通过大量的实例，介绍在可视环境下进行面向对象程序设计的方法、步骤，力求通过实际操作让读者熟悉 Visual FoxPro 的使用方法。通过一个个具有针对性的实例和具体的操作过程，让读者很快熟悉 Visual FoxPro 的应用环境，从而实现从易到难，边学、边练、边提高的学习过程。

本书以 Visual FoxPro 6.0 中文版为平台，结合普通高校非计算机专业数据库程序设计课程的具体要求，深入浅出地介绍了 Visual FoxPro 数据库程序设计的有关知识、方法和具体的实例。本教材共分 10 章，分别介绍了数据库系统及 Visual FoxPro 的基本概念、项目管理器及其操作、数据表的基本操作、数据库的设计与操作、程序设计基础、面向对象程序设计、表单的建立与使用、结构化查询语言（SQL）、查询与视图、报表设计、菜单设计、应用程序的生成和发布等。

本书以基础知识、基本理论和基本方法为着眼点，努力做到知识体系完整，结构顺序合理，内容深度适宜，例题典型全面，讲解深入浅出，使用起来得心应手。在最后一章中给出了应用系统开发的具体案例，通过该案例可使读者更加详细地了解 and 掌握应用系统开发的一般步骤、基本方法和具体过程，使读者快速获得应用系统开发的基本技能。

本书的风格简洁实用，内容贴近实际应用，是一本较好地将学习与实践结合在一起的书籍。本书内容翔实、案例完整、步骤详尽，具有很强的实用性，十分适合那些希望尽快实现应用开发的读者的需要。

由于作者水平有限，加上编写时间仓促，错误和不足之处在所难免，敬请广大读者批评指正。

# 目 录

第 1 章 Visual FoxPro 基础知识.....	(1)
1.1 数据库基础概念.....	(1)
1.1.1 数据库相关知识.....	(1)
1.1.2 数据模型.....	(2)
1.1.3 关系模型的基本概念.....	(4)
1.1.4 关系运算.....	(6)
1.1.5 实体模型.....	(7)
1.2 Visual FoxPro 6.0 及其界面.....	(8)
1.2.1 Visual FoxPro 6.0 概述.....	(8)
1.2.2 Visual FoxPro 6.0 的启动与退出.....	(9)
1.2.3 Visual FoxPro 6.0 系统的主窗口.....	(10)
1.3 项目管理器.....	(12)
1.3.1 【项目管理器】窗口的组成.....	(12)
1.3.2 建立项目文件.....	(13)
1.3.3 使用项目管理器管理项目文件.....	(14)
1.3.4 操作【项目管理器】窗口.....	(15)
1.3.5 Visual FoxPro 6.0 的辅助设计工具.....	(16)
1.4 学生成绩管理系统案例分析.....	(17)
1.4.1 案例设计分析.....	(17)
1.4.2 学生成绩管理系统功能模块设计.....	(18)
1.4.3 学生成绩管理系统设计要求.....	(23)
思考与练习 1.....	(23)
第 2 章 数据与数据运算.....	(26)
2.1 数据与运算.....	(26)
2.1.1 数据类型.....	(26)
2.1.2 常量和变量.....	(26)
2.1.3 运算符及表达式.....	(28)
2.2 常用函数.....	(31)
2.2.1 数学运算函数.....	(32)
2.2.2 字符处理函数.....	(33)
2.2.3 日期和时间函数.....	(35)
2.2.4 测试函数.....	(35)



2.2.5	转换函数.....	(37)
2.2.6	用户自定义对话框 MESSAGEBOX()函数的使用.....	(38)
	思考与练习 2.....	(41)
<b>第 3 章</b>	<b>数据库的基本操作.....</b>	<b>(43)</b>
3.1	数据库与数据表.....	(43)
3.1.1	数据库的基本操作.....	(43)
3.1.2	数据表的基本概念.....	(45)
3.1.3	数据表结构的建立.....	(47)
3.1.4	数据表结构的操作.....	(50)
3.1.5	数据表记录的操作.....	(53)
3.2	数据表的索引与筛选.....	(62)
3.2.1	建立索引.....	(62)
3.2.2	筛选记录.....	(65)
3.3	Visual FoxPro 中的命令操作.....	(66)
3.3.1	Visual FoxPro 命令操作的基本概念.....	(66)
3.3.2	数据库、数据表的操作命令.....	(70)
3.3.3	表记录的操作命令.....	(71)
3.3.4	追加记录命令.....	(73)
3.3.5	记录的排序和索引.....	(74)
3.3.6	记录的查询.....	(75)
3.3.7	数值统计.....	(76)
3.4	多表操作.....	(78)
3.4.1	数据表的关联.....	(78)
3.4.2	设置参照完整性.....	(80)
	思考与练习 3.....	(83)
<b>第 4 章</b>	<b>查询与视图.....</b>	<b>(85)</b>
4.1	查询.....	(85)
4.1.1	查询文件的建立与修改.....	(85)
4.1.2	查询文件的执行.....	(91)
4.2	视图.....	(98)
4.2.1	利用向导创建本地视图.....	(99)
4.2.2	利用视图设计器创建本地视图.....	(100)
4.2.3	创建参数化视图.....	(104)
4.2.4	创建远程视图.....	(105)
4.3	关系数据库标准语言 SQL.....	(108)
4.3.1	SQL 概述.....	(108)
4.3.2	SQL 的各类查询.....	(110)
4.3.3	操作功能.....	(125)

4.3.4	定义功能	(126)
4.3.5	定义视图	(130)
	思考与练习 4	(134)
<b>第 5 章</b>	<b>Visual FoxPro 程序设计基础</b>	<b>(136)</b>
5.1	程序设计基础	(136)
5.1.1	程序的建立与维护	(136)
5.1.2	编程基础	(138)
5.2	程序的流程控制	(141)
5.2.1	选择结构	(141)
5.2.2	循环结构	(146)
5.3	模块结构程序设计	(151)
5.3.1	过程及过程调用	(151)
5.3.2	过程调用中的参数传递	(154)
5.3.3	变量的作用域	(156)
5.3.4	自定义函数	(157)
	思考与练习 5	(161)
<b>第 6 章</b>	<b>表单</b>	<b>(165)</b>
6.1	面向对象程序设计简介	(165)
6.1.1	面向对象程序设计中的基本概念	(165)
6.1.2	Visual FoxPro 中的类	(166)
6.1.3	Visual FoxPro 中的对象	(168)
6.1.4	Visual FoxPro 中的事件和事件过程	(171)
6.2	表单的创建	(174)
6.2.1	使用表单向导创建表单	(174)
6.2.2	使用表单设计器创建表单	(178)
6.2.3	表单控件	(181)
6.3	常用控件的使用方法	(185)
6.3.1	标签、文本框和命令按钮(组)的使用	(185)
6.3.2	编辑框、列表框和组合框的使用	(190)
6.3.3	选项按钮组、复选框和微调按钮的使用	(192)
6.3.4	表格和计时器的使用	(196)
6.3.5	OLE 绑定控件和图像的使用	(198)
	思考与练习 6	(206)
<b>第 7 章</b>	<b>菜单与工具栏</b>	<b>(208)</b>
7.1	菜单的建立	(208)
7.1.1	菜单概述	(208)
7.1.2	创建快速菜单	(211)

7.1.3	创建菜单	(212)
7.1.4	快捷菜单的建立	(218)
7.1.5	设置对菜单项的响应	(219)
7.2	工具栏	(224)
7.2.1	定制与建立工具栏	(224)
7.2.2	给新工具栏命名	(226)
7.2.3	定义对象操作	(226)
	思考与练习 7	(228)
<b>第 8 章</b>	<b>报表与标签</b>	<b>(230)</b>
8.1	报表的设计	(230)
8.1.1	报表类型	(230)
8.1.2	利用报表向导创建报表	(230)
8.1.3	使用设计器创建报表	(235)
8.1.4	对报表进行细节设计	(237)
8.1.5	完善报表	(242)
8.2	创建标签	(245)
8.2.1	使用标签向导创建标签	(245)
8.2.2	利用标签设计器创建标签	(247)
8.3	报表和标签的输出	(249)
8.3.1	页面设置	(249)
8.3.2	打印输出报表和标签	(250)
	思考与练习 8	(251)
<b>第 9 章</b>	<b>应用程序的编译与发布</b>	<b>(253)</b>
9.1	应用程序的开发	(253)
9.1.1	系统开发步骤	(253)
9.1.2	连编应用程序	(254)
9.1.3	主程序设计	(256)
9.2	应用程序生成器	(259)
9.2.1	使用应用程序向导	(259)
9.2.2	应用程序生成器	(260)
9.2.3	建立 Visual FoxPro 应用程序	(262)
9.2.4	制作安装盘	(266)
	思考与练习 9	(270)
<b>第 10 章</b>	<b>小型系统项目开发</b>	<b>(272)</b>
10.1	教职工信息管理系统	(272)
10.1.1	案例设计的提出及要解决的问题	(272)
10.1.2	案例设计要实现的主要功能	(272)

10.1.3 案例设计操作思路 ..... (273)

10.1.4 案例设计参考步骤 ..... (273)

10.2 图书信息管理系统..... (285)

10.2.1 案例设计的提出及要解决的问题..... (285)

10.2.2 案例设计要实现的主要功能 ..... (285)

10.2.3 案例设计操作思路 ..... (285)

10.2.4 案例设计参考步骤 ..... (285)

思考与练习 10 ..... (299)

参考文献..... (301)

# 第 1 章 Visual FoxPro 基础知识

## 1.1 数据库基础概念

数据库管理技术是信息科学的重要组成部分。随着商品经济的发展、科学技术的进步和激烈的市场竞争，社会信息量倍增，决策难度也随之加大，使得计算机处理的数据量不断增加。于是数据库管理系统便应运而生，从而也促进了信息科学的发展。下面我们从数据库基础概念开始介绍。

### 1.1.1 数据库相关知识

#### 1. 数据、信息

① 数据 (data): 是对客观事物特征所进行的一种抽象化、符号化的表示。通俗地讲，凡是能被计算机接受，并能被计算机处理的数字、字符、图形、声音、图像等统称为数据。数据所反映的事物属性是它的内容，而符号是它的形式。

② 信息 (information): 是客观事物属性的反映。它所反映的是关于某一客观系统中某一事物的某一方面属性或某一时刻的表现形式。通俗地讲，信息是经过加工处理并对人类客观行为产生影响的数据表现形式。也可以说，信息是有一定含义的并经过加工处理，能够提供决策性依据的数据。

#### 2. 数据处理

所谓数据处理实际上就是利用计算机对各种类型的数据进行处理。它包括对数据的采集、整理、存储、分类、排序、检索、维护、加工、统计和传输等一系列操作过程。数据处理的目的是从大量的、原始的数据中获得我们所需要的资料并提取有用的数据成分，作为行为和决策的依据。

#### 3. 数据库

数据库在英语中称为 database。分解这个单词来看，data 是数据，base 可译为基地或仓库。所以在通俗的意义上，不妨将数据库理解为存储数据的基地。它是以一定的组织方式将相关的数据组织在一起并存储在外部存储器上，所形成的能为多个用户共享的、与应用程序彼此独立的一组相互关联的数据集合。

#### 4. 数据库管理系统

数据库管理系统简称 DBMS (Data Base Management System)。DBMS 是操纵和管理数据库的软件，是数据库系统的管理控制中心，一般有 4 大功能：数据定义功能、数据库操作功能、控制和管理功能、建立和维护功能。

5. 数据库系统

把以数据库应用为基础的计算机系统称为数据库系统。它是一个实际可行的并按照数据库方式存储、维护和管理数据的系统。数据库系统通常由计算机硬件、数据库、数据库管理系统、相关软件、人员（数据库管理员、应用程序员、用户）等组成，如图 1.1 所示。

6. 数据库应用系统

数据库应用系统是一个复杂的系统，它由硬件、操作系统、数据库管理系统、编译系统、用户应用程序和数据库组成。

数据库、数据库管理系统和数据库系统是 3 个不同的概念，数据库管理系统在计算机中的地位如图 1.2 所示。

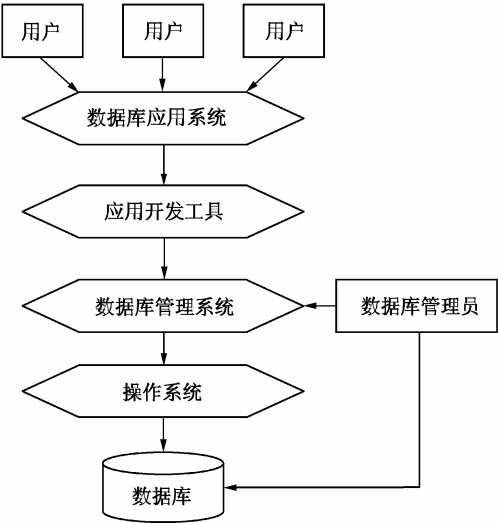


图 1.1 数据库系统的组成结构图

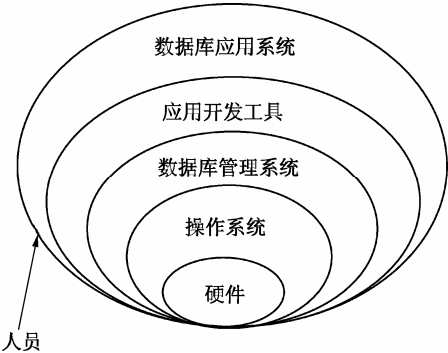


图 1.2 数据库管理系统在计算机中的地位

1.1.2 数据模型

数据模型是数据库系统的核心，也是定义数据库模型的基础，其好坏直接影响数据库的性能。

现实世界中的客观事物是彼此相互联系的。一方面，某一事物内部的诸因素和属性根据一定的组织原则相互具有联系，构成一个相对独立的系统；另一方面，某一事物同时也作为一个更大系统的一个因素或一种属性而存在，并与系统的其他因素或属性发生联系。客观事物的这种普遍联系性决定了作为事物属性记录符号的数据与数据之间也存在着一定的联系性。具有联系性的相关数据总是按照一定的组织关系排列，从而构成一定的结构，对这种结构的描述就是数据模型。

从理论上讲，数据模型是指反映客观事物及客观事物之间联系的数据组织的结构和形式。客观事物是千变万化的，各种客观事物的数据模型也是千差万别的，但也有其共同性。常用的数据模型有层次模型、网状模型和关系模型三种。

1. 层次模型

层次模型（hierarchical model）表示数据间的从属关系结构，是一种以记录某一事物的类型为根节点的有向树结构。层次模型像一棵倒置的树，根节点在上，层次最高；子节点在下，逐层排列。这种用树形结构表示数据之间联系的模型也称为树结构。层次模型的特点是：仅有一个无双亲的根节点；根节点以外的子节点，向上仅有一个父节点，向下有若干子节点。

层次模型表示的是从根节点到子节点的一个节点对多个节点，或从子节点到父节点的多个节点对一个节点的数据间的联系，如图 1.3 所示。

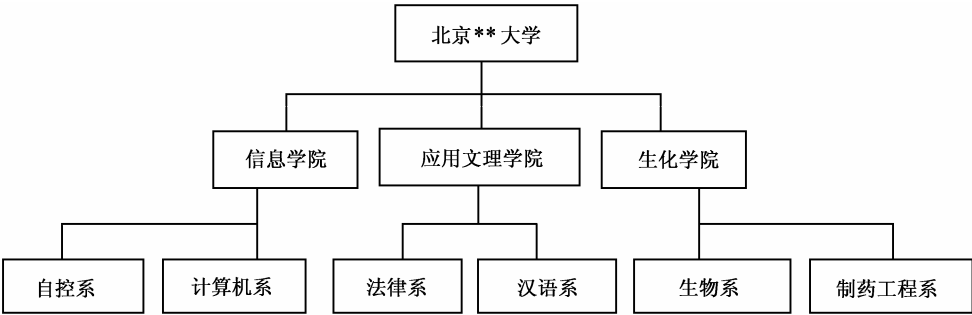


图 1.3 层次模型

2. 网状模型

网状模型（network model）是层次模型的扩展，表示多个从属关系的层次结构，呈现一种交叉关系的网络结构。网状模型是以记录为节点的网络结构，用网状数据结构表示实体与实体之间联系。网状模型的特点是：可以有一个以上的节点无双亲，至少有一个节点有多于一个的双亲。因此，层次模型是网状模型的特殊形式，网状模型可以表示较复杂的数据结构，即可以表示数据间的纵向关系与横向关系。这种数据模型在概念上、结构上都比较复杂，在操作上也有很多不便，如图 1.4 所示。

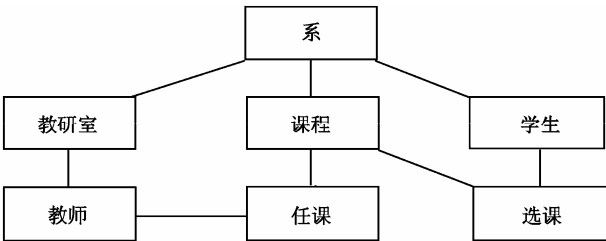


图 1.4 网状模型

3. 关系模型

关系模型（relational model）的所谓“关系”是有特定含义的。广义地说，任何数据模型都描述一定事物数据之间的关系。关系中的每一个数据项也称“字段”是不可再分的，它是最基本的单位；并且每一竖列数据项的属性是相同的。列数根据需要而设，并且各列的顺

序是任意的；每一横行记录由一个事物的诸多属性项构成。记录的顺序可以是任意的；一个关系是一张二维表，不允许有相同的字段名，也不允许有相同的记录行。

关系数据库采用人们经常使用的表格作为基本的数据结构，通过公共的关键字字段来实现不同二维表之间（或“关系”之间）的数据联系。可见关系模型呈二维表形式，如表 1.1 所示，这种结构简单明了，使用与学习起来都很方便（表中的“学号”、“姓名”等即为字段名）。

表 1.1 学生表

学号	姓名	专业编号	性别	出生日期	入学成绩	团员否	照片	简历
080301001	张跃林	03	男	1989-10-23	589	T	—	—
080301020	张文斌	03	男	1989-5-20	593	T	—	—
080302045	陈江城	03	男	1989-8-4	598	T	—	—
084201002	夏利华	42	女	1989-9-4	497	F	—	—
074202123	李林萍	42	女	1989-3-5	516	T	—	—
...	...	...	...	...	...	...	...	...

1.1.3 关系模型的基本概念

关系数据库（relation database）是若干个依照关系模型设计的数据表文件的集合。也就是说，关系数据库是由若干张完成关系模型设计的二维表组成的。一张二维表为一个数据表，数据表包含数据及数据间的关系。

一个关系数据库由若干个数据表组成，数据表又由若干个记录组成，而每一个记录是由若干个以字段属性加以分类的数据项组成的。图 1.5 中的教师表就是一个关系模型，它包含以下的概念。

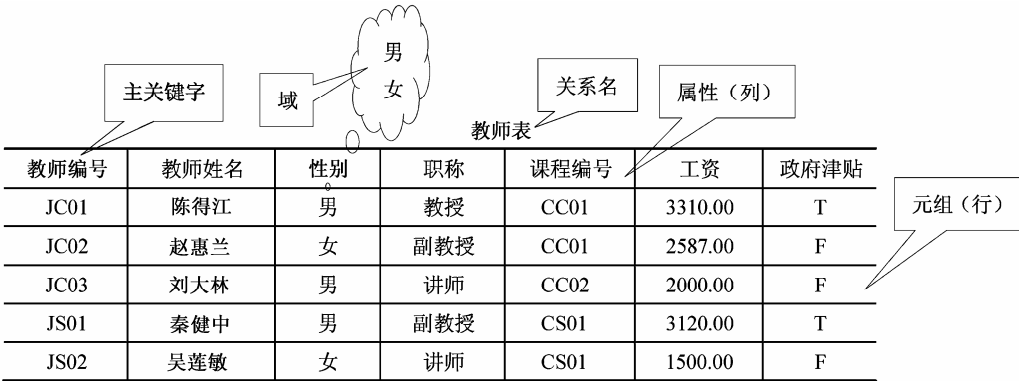


图 1.5 教师表

1. 关系

通常将一个没有重复行、重复列的二维表看成一个关系。一个关系就是一张二维表，每个关系有一个关系名。在 Visual FoxPro 中，一个关系即为一个表文件，其扩展名为 .dbf。

在 Visual FoxPro 中，一个关系对应于一个表文件，关系名则对应于表文件名或表名。



## 2. 元组

二维表的每一行在关系中称为元组。在 Visual FoxPro 中，一个元组对应表中的一个记录。

## 3. 属性

二维表的每一列在关系中称为属性，每个属性都有一个属性名，属性值则是各个元组属性的取值。在 Visual FoxPro 中，一个属性对应表中的一个字段，属性名对应字段名，属性值对应各个记录的字段值。

## 4. 域

属性的取值范围称为域。域作为属性值的集合，其类型与范围具体由属性的性质及其所表示的意义确定。如图1.5中“性别”属性的域是{男，女}。同一属性只能在相同域中取值。

## 5. 关键字

关系中能唯一区分、确定不同元组的属性或属性组合称为该关系的一个关键字。单个属性组成的关键字称为单关键字，多个属性组成的关键字称为组合关键字。需要强调的是，关键字的属性值不能取“空值”，因为无法唯一区分、确定元组。

## 6. 候选关键字

关系中能够成为关键字的属性或属性组合可能不是唯一的。凡在关系中能够唯一区分、确定不同元组的属性或属性组合都称为候选关键字。

## 7. 主关键字

在候选关键字中选定一个作为关键字，称为该关系的主关键字。在关系中，主关键字是唯一的。

## 8. 外部关键字

关系中某个属性或属性组合并非关键字，但却是另一个关系的主关键字，称此属性或属性组合为本关系的外部关键字。关系之间的联系是通过外部关键字实现的。外部关键字用来实现表与表之间的关联，实现表与表之间的参照完整性。

## 9. 关系模式

对关系的描述称为关系模式，其格式为

关系名(属性名 1, 属性名 2, ..., 属性名  $n$ )

关系既可以用二维表格描述，也可以用数学形式的关系模式来描述。一个关系模式对应一个关系的数据结构，也就是表的数据结构。

1.1.4 关系运算

关系的基本运算有两类：一类是传统的集合运算，包括并、差、交；另一类是专门的关系运算，包括选择、投影和连接。

1. 传统的集合运算

进行并、差、交集集合运算的两个关系必须具有相同的结构模式，即结构相同。表 1.2 和表 1.3 给出了两种关系，用于介绍接下来的集合运算。

表 1.2 R 关系

学号	姓名	性别
080301001	张跃林	男
080301020	张文斌	男
084201002	夏利华	女

表 1.3 S 关系

学号	姓名	性别
084205255	李雅迪	女
080401081	孟江浩	男
084201002	夏利华	女

① 并：两个相同结构关系的并是由属于这两个关系的元组（记录）组成的集合，如表 1.4 所示。

② 差：有关系 R 和关系 S，是由属于 R 而不属于 S 的元组组成的集合，从 R 中去掉 S 中也含有的元组，如表 1.5 所示。

③ 交：有关系 R 和关系 S，是既属于 R 又属于 S 的元组组成的集合，如表 1.6 所示。

表 1.4 R ∪ S

学号	姓名	性别
080301001	张跃林	男
080301020	张文斌	男
084201002	夏利华	女
084205255	李雅迪	女
080401081	孟江浩	男

表 1.5 R - S

学号	姓名	性别
080301001	张跃林	男
080301020	张文斌	男

表 1.6 R ∩ S

学号	姓名	性别
084201002	夏利华	女

2. 专门的关系运算

在关系数据库中查询用户所需的数据时，需要对关系进行一定的关系运算。关系运算主要有选择、投影和连接三种。

① 选择：从关系中找出满足条件的记录。是一种横向的操作，它可以根据用户的要求从关系中筛选出满足一定条件的记录，通过这种运算可以得到一个新的关系，但其中的元组是原关系的一个子集，不会影响关系的结构。例如 Visual FoxPro 中的“FOR<条件>”命令、“WHILE<条件>”命令等。

② 投影：从关系中选取若干属性组成新的关系。是一种列的操作，它可以根据用户的要求从关系选出若干个字段组成新的关系，字段的个数或顺序往往不同。例如 Visual FoxPro 中的“FIELDS<字段 1, 字段 2, 字段 3, …>”命令等。

③ 连接：将两个关系通过公共属性名连接成一个新的关系。连接运算可以实现两个关系的横向合并，在新的关系中可以反映出原来关系之间的联系。在连接运算中，按照字段值对应相等为条件进行的连接操作称为等值连接。自然连接是去掉重复属性的等值连接。

### 3. 关系数据库

关系数据库是由若干个依照关系模型设计的二维数据表文件的集合。一个关系数据库即为一个数据库文件。

### 4. 关系的完整性

关系的完整性是为保证数据库中数据的正确性和兼容性对关系模型提出的某种约束条件或规则。完整性通常包括实体完整性、参照完整性和域完整性，其中实体完整性和参照完整性是关系模型必须满足的完整性约束条件。

① 实体完整性：实体完整性是指关系的主关键字不能取“空值”。

② 参照完整性：参照完整性是定义建立关系之间联系的主关键字与外部关键字引用的约束条件。如“选课表”关系中课程号的值只能取“课程表”关系中已存在课程号的值。

③ 域完整性：实体完整性和参照完整性适用于任何关系型数据库系统，主要是对关系的主关键字和外部关键字取值必须做出有效的约束。域完整性则是根据应用环境的要求和实际需要，对某一具体应用所涉及的数据提出约束性条件。这一约束机制一般不应由应用程序提供，而应由关系模型提供定义并检验。如检验属性的类型、宽度等，进一步保证输入数据合理有效。域完整性主要包括字段有效性约束和记录有效性约束。

## 1.1.5 实体模型

### 1. 实体

客观存在并且可以相互区别的事物称为实体。实体可以是具体的，例如一个学生、一本书、一名教师；也可以是抽象的，例如一堂课、一次足球比赛。

### 2. 属性

描述实体的特征称为该实体的属性。例如，学生有学号、姓名、专业编号、性别、出生日期、入学成绩等方面的属性。属性有“型”和“值”之分，型即为属性名，值即为属性的具体内容（例如“084201002”、“夏利华”、“42”、“女”、“1989-9-4”、“497”、“F”）。

### 3. 实体间的联系

实体之间的对应关系称为联系，它反映现实世界事物之间的相互关联。例如，学生和课程是两个不同的实体，当学生选课时，两者之间则发生了关联，建立了联系——学生选择课程，课程被学生选学。实体间的联系分为两种：一种是实体内部各属性之间的联系，另一种是实体之间的联系。

### 1) 一对一联系 (1:1)

实体集 A 中的一个实体至多与实体集 B 中的一个实体相对应；反之，实体集 B 中的一个实体至多对应于实体集 A 中的一个实体，则称实体集 A 与实体集 B 为一对一联系。例如，厂长与工厂之间、观众与座位之间、旅客与车票之间、病人与病床之间等。

### 2) 一对多联系 (1:N)

实体集 A 中的一个实体与实体集 B 中的  $N$  ( $N \geq 0$ ) 个实体相对应；反之，实体集 B 中的一个实体至多与实体集 A 中的一个实体相对应。例如，学校与系、班级与学生、省与市等。

### 3) 多对多联系 (M:N)

实体集 A 中的一个实体与实体集 B 中的  $N$  ( $N \geq 0$ ) 个实体相对应；反之，实体集 B 中的一个实体与实体集 A 中的  $M$  ( $M \geq 0$ ) 个实体相对应。例如，教师与学生、学生与课程、工厂与产品、商店与顾客等。

## 练习与操作

练习 1 指出表 1.1 的“学生表”属于什么关系模型。

练习 2 指出表 1.1 的“学生表”中的“关系名”、“主关键字”、“属性”、“元组”、“域”。

练习 3 回答并完成下列问题：

- (1) 数据与信息的异同是什么？
- (2) 实体间联系的种类有哪些？
- (3) 层次模型、网状模型和关系模型的区别是什么？
- (4) 数据库能够把大量数据按一定的结构进行存储，\_\_\_\_\_，并实现数据共享。
- (5) 表由若干条记录组成，每一行称为一个\_\_\_\_\_，对应着一个真实对象的每一列称为一个“字段”。

## 1.2 Visual FoxPro 6.0 及其界面

### 1.2.1 Visual FoxPro 6.0 概述

Visual FoxPro 6.0 中文版作为一种关系型数据库系统，不仅简化了数据库的管理，在应用程序或数据库开发的任何领域都提供了帮助。而且它既支持编程方式又支持交互方式，使普通用户能方便地利用可视化的设计工具和向导来快速创建表、查询、视图、报表和表单等。Visual FoxPro 6.0 增强了以下的性能。

① 增强了项目和数据库管理能力。在 Visual FoxPro 6.0 中，对项目管理器和数据库设计器进行了改进，可以方便地管理项目和数据库。

② 改进了程序调试工具。在 Visual FoxPro 6.0 中，有一个更为简单的调试和修改程序，可以在主窗口打开调试工具，与应用程序界面一起显示；使用代码范围分析程序可以分析运行的程序代码。

③ 提供了更简单的表设计。在 Visual FoxPro 6.0 中，用户可以非常方便地利用表设计器来创建表，可以在创建表字段的同时方便地添加索引，指定多种默认值。通过表设计器中的选项卡，可以直接设定表的有效规则、触发器及统计值。

④ 增强了查询和视图设计功能。在 Visual FoxPro 6.0 中,使用查询设计器创建和修改查询,使用视图设计器创建和修改视图,用户既可以利用表之间已有的关系作为内部连接,也可以创建外部连接,为列指定别名等。

⑤ 增强了表单设计功能。表单及表单控件在原有的基础上增加了一些属性和方法,使其能更好地使用表单及其控件,在表单设计器的属性窗口可以选择多个控件进行查看并更改它们的共同的属性。

⑥ 提供了更多、更好的向导。Visual FoxPro 6.0 中的向导和实例提供了更强大的功能。表单向导允许添加一个页框控件,从而显示更多的字段。

⑦ 兼容性好。使用 dBASE 或 FoxPro 的早期版本开发的程序,几乎不用修改就可在 Visual FoxPro 6.0 中运行。

⑧ 应用程序开发更方便。Visual FoxPro 6.0 新增了“应用程序向导”,可以更有效地开发应用程序。

⑨ 全新的网络功能。Visual FoxPro 6.0 还提供了一个 Web 页面的搜索向导,用以创建查询或显示数据库记录的 Web 页面。

## 1.2.2 Visual FoxPro 6.0 的启动与退出

### 1. Visual FoxPro 6.0 的启动

在 Windows XP 操作系统中安装了 Visual FoxPro 6.0 之后,安装程序会在【开始】菜单的【所有程序】中增加一个新的级联菜单“Microsoft Visual FoxPro 6.0”,即在 C:\Program Files\Microsoft Visual Studio\VFP98 的目录下增加一个快捷方式。要启动 Visual FoxPro 6.0,可按下述几种方式进行操作。

① 双击桌面上的 Visual FoxPro 6.0 图标,直接进入 Visual FoxPro 6.0。

② 单击 Windows 桌面的【开始】→【运行】命令,输入打开路径 C:\Programs Files\Microsoft Visual Studio\VFP98\VFP6.0.exe,然后单击【确定】按钮进入 Visual FoxPro 6.0。

③ 单击 Windows 桌面的【开始】→【所有程序】→【Microsoft Visual FoxPro 6.0】→【Microsoft Visual FoxPro 6.0】命令,可以启动 Visual FoxPro 6.0。

④ 此外,也可以在 Windows 系统的“资源管理器”中找到“VFP6.0.exe”文件,然后双击该文件名进入 Visual FoxPro 6.0。

在刚启动 Visual FoxPro 6.0 时看到的一个画面称为启动界面,如图 1.6 所示。其中显示了 Visual FoxPro 6.0 的商标、版本号及其他信息。

每当用户启动 Visual FoxPro 6.0 系统时,一开始便会出现启动界面,在欢迎界面中只要单击【关闭此屏】选项就可进入 Visual FoxPro 6.0 系统。如果希望以后不再出现此画面,可在【以后不再显示此屏】选项前打上“√”即可。

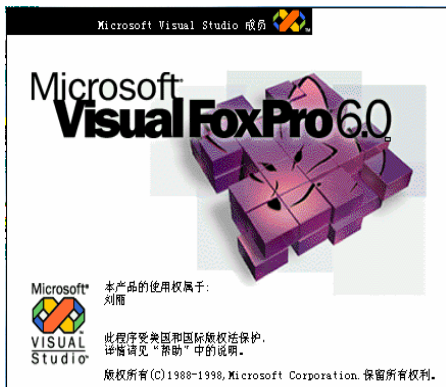



图 1.6 启动界面

## 2. Visual FoxPro 6.0 的退出

在 Visual FoxPro 6.0 中对数据库或表进行操作后, 为了保证数据的安全, 必须正确退出 Visual FoxPro 6.0。在退出 Visual FoxPro 6.0 之前, 应该将所有打开的表全部关闭。可用以下任何一种方法退出 Visual FoxPro 6.0。

- ① 使用鼠标单击 Visual FoxPro 6.0 窗口右上角的关闭按钮 (✕)。
- ② 选择【文件】菜单中的【退出】命令。
- ③ 在【命令】窗口输入命令“Quit”, 然后按下 Enter 键。
- ④ 直接按下 Alt+F4 组合键。
- ⑤ 双击 Visual FoxPro 6.0 主窗口左上角的  控制按钮图标。

在使用 Visual FoxPro 6.0 的过程中, 如果直接关闭主机电源、热启动或按下复位键, 都有可能造成用户数据丢失。

### 1.2.3 Visual FoxPro 6.0 系统的主窗口

进入 Visual FoxPro 6.0 之后, 计算机桌面上显示的就是 Visual FoxPro 6.0 系统的主窗口, 如图 1.7 所示。在 Visual FoxPro 6.0 系统的主窗口中可以看到标题栏、菜单栏、工具栏、状态栏、【命令】窗口和【项目管理器】窗口等组成项目。

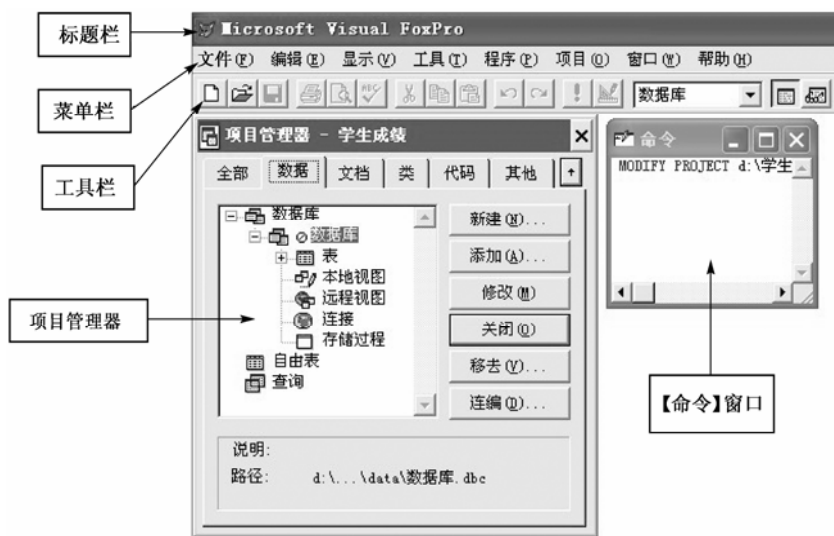


图 1.7 Visual FoxPro 6.0 系统的主窗口

#### 1. 标题栏

标题栏位于主窗口的顶部, 包含主窗口的控制菜单图标、应用程序名, 以及右边的最小化按钮、最大化按钮或还原按钮、关闭按钮。

#### 2. 菜单栏

Visual FoxPro 6.0 的系统菜单栏位于标题栏的下方, 默认情况下有 8 个系统菜单项。随

着用户操作的不同，系统菜单中会动态地显示一些其他的菜单，使用系统菜单能实现数据库管理和应用程序开发的大部分操作。系统菜单是当前可用的命令的集合，如图 1.8 所示。系统菜单可以完成约 70 条命令。

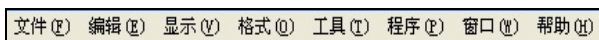


图 1.8 Visual FoxPro 6.0 的菜单栏

需要指出的是，系统菜单的内容并非一成不变的。当在 Visual FoxPro 6.0 中操作不同的对象时，系统菜单的内容也会随之而发生改变。如图 1.9 所示为打开项目管理器时的菜单栏，可以与图 1.8 比较二者之间的区别。

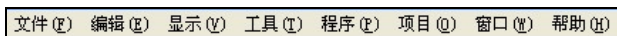


图 1.9 打开项目管理器时的菜单栏

### 3. 工具栏

工具栏位于菜单栏的下方。Visual FoxPro 6.0 有多个工具栏，可根据操作需要进行添加和设置。默认情况下，Visual FoxPro 6.0 工具栏显示在屏幕顶部的菜单栏下面。实际上，工具栏中的每一个按钮在菜单中都有选项与之对应。在【工具栏】对话框中，用户可以根据自己的需要进行工具栏的定制、工具栏的删除等操作。Visual FoxPro 6.0 中提供了 11 个预设的工具栏，第一次启动 Visual FoxPro 6.0 时只有“常用”工具栏显示出来。如果要显示其他工具栏，可选择系统菜单中的【显示】→【工具栏】命令，出现如图 1.10 所示的【工具栏】对话框。在该对话框中可根据需要选择工具栏中的工具。也可在工具栏区域单击鼠标右键，在出现的快捷菜单中进行选择，如图 1.11 所示。

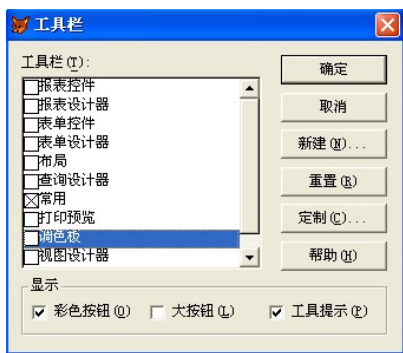


图 1.10 【工具栏】对话框

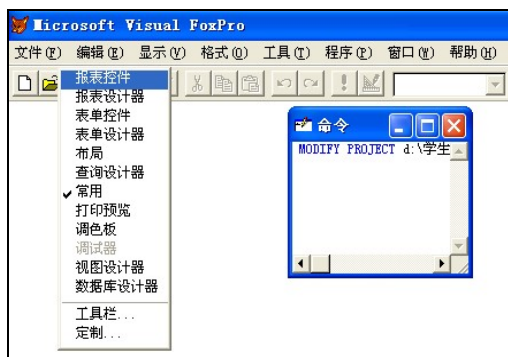


图 1.11 在工具栏区域单击鼠标右键出现的快捷菜单

### 4. 【命令】窗口

【命令】窗口是一个非常有用的窗口。在【命令】窗口中初学者可以学习 Visual FoxPro 6.0 的常用命令和函数的用法。另外，当使用菜单或工具栏进行操作时，相应的命令也会自动显示在【命令】窗口中。【命令】窗口具有完整的 Windows 风格，可以对它进行移动、放大、缩小、最大化等，其操纵与普通的 Windows 窗口一样。

## 5. 状态栏

状态栏用于显示当前的操作状态，各种菜单、控件的功能说明，以及命令提示、时钟等信息。

## 练习与操作

### 练习 1 安装 Visual FoxPro 6.0。

插入 Visual FoxPro 6.0 光盘，光盘上的 AutoRun 程序自动运行（如果不是专门的 Visual FoxPro 光盘，可以从【我的电脑】或【资源管理器】中选择应用程序后运行），根据中文提示逐步安装，安装时最好安装到 D: 盘上。系统会自动建立一个文件夹 Visual FoxPro，并且把所有系统文件装入其中。安装完成后，重新启动 Windows 操作系统，随后系统将完成相关的参数设置，即可启动 Visual FoxPro 6.0。

### 练习 2 启动 Visual FoxPro 6.0。

- (1) 直接双击桌面的 Visual FoxPro 6.0 快捷图标（狐狸图标）。
- (2) 单击 Windows 桌面上的【开始】→【所有程序】→【Microsoft Visual FoxPro 6.0】→【Microsoft Visual FoxPro 6.0（狐狸图标）】命令启动后，屏幕上出现图 1.7 所示的主窗口。

### 练习 3 退出 Visual FoxPro 6.0。

- (1) 在【命令】窗口中，输入“QUIT”。
- (2) 按下 Alt + F4 组合键。
- (3) 在【文件】菜单中，选择【退出】命令。
- (4) 单击主窗口左上角的“狐狸”图标，选择【关闭】命令。

### 练习 4 自定义工具栏。

- (1) 选择菜单栏上的【显示】→【工具栏】命令，选定工具栏后，单击【确定】按钮，如图 1.10 所示。
- (2) 在工具栏的任一空白处单击鼠标右键，在快捷菜单中选定所需工具，如图 1.11 所示。

## 1.3 项目管理器

### 1.3.1 【项目管理器】窗口的组成

通过项目管理器，可以很快地熟悉 Visual FoxPro 的各种对象。项目管理器可以将数据库系统中的各类文件集中管理，并编译成应用软件。编译后即可为可执行的应用程序文件，可形成快捷键，可直接启动数据库应用软件，操作十分方便。通过项目管理器可以实现对数据库、数据表、表单、查询、视图、报表、程序及相关文件的管理。【项目管理器】窗口如图 1.12 所示。

#### 1. 选项卡

【项目管理器】窗口由 6 个选项卡组成，各选项卡的功能如下。



- **【全部】**：显示和管理项目包含的所有文件。
- **【数据】**：负责管理项目的数据库、自由表、查询及视图等内容。
- **【文档】**：负责管理处理数据时的全部文档，如输入和查看数据所用的表单，以及打印表和查询选项卡结果所用的报表及标签等。
- **【类】**：包含用户所有定义的类，可以新建一个类，也可以添加已存在的类。
- **【代码】**：包含用户创建的所有源程序（.api 文件）和 API 库。
- **【其他】**：包含用户创建的菜单、文本文件及其他文件。用户可以设计自己的菜单栏及快捷菜单。文本文件是存储纯文本的文件，可以用来存储文件的说明信息。其他文件用来存储图形文件，例如“.bmp”、“.jpg”格式的文件。

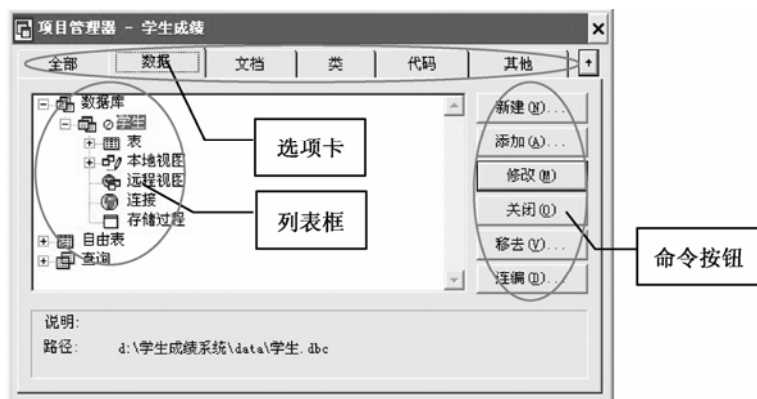


图 1.12 【项目管理器】窗口

## 2. 命令按钮

在【项目管理器】窗口的右边有几个命令按钮，选择要操作的某一文件，再单击相应的命令按钮即可进行相关的操作。常见的命令按钮有：**【新建】**、**【添加】**、**【修改】**、**【关闭】**、**【移去】**、**【连编】**等。

## 3. 列表框

列表框位于【项目管理器】窗口的左边，用于显示选定的选项卡下的所有文件列表。

### 1.3.2 建立项目文件

现在新建一个项目，从而管理自己的应用程序。在【文件】菜单中选择**【新建】**命令，进入【新建】对话框（如图 1.13 所示），在对话框中选择**【项目】**选项。

在【新建】对话框的右侧有两个按钮，单击**【新建文件】**按钮，可以通过系统的设计器来创建项目；也可单击**【向导】**按钮，由系统提供的向导来创建项目。选择单击**【新建文件】**



图 1.13 【新建】对话框

按钮,这时可看到【创建】对话框,在其中输入文件名、新建文件的存储路径之后,单击【保存】按钮。这样就生成一个空的项目文件。

### 1.3.3 使用项目管理器管理项目文件

新建的项目文件是一个空文件,用户只有将自己的资源添加到该项目文件中,才能有效地使用项目管理器进行管理。

#### 1. 向项目管理器中加入一个新文件

如果在 Visual FoxPro 6.0 已经存在多个对象文件,可以将这些文件添加到项目管理器中,例如向项目中添加查询等。

在【项目管理器】窗口中选择想要加入的文件类型(如“表”),单击【项目管理器】窗口中的【添加】按钮,在打开的对话框中选择表的名字和表所在的路径,单击【确定】按钮完成添加操作,如图 1.14 所示。

#### 2. 从项目管理器中移去文件

对于一些不需要的文件,可以将这些文件从项目管理器中移去,例如移去项目中的数据库等。可以在【项目管理器】窗口中选择想要移去的文件,单击【项目管理器】窗口中的【移去】按钮,并在随后显示的对话框中单击【移去】按钮即可;如果想从计算机中删除文件,则单击【删除】按钮,如图 1.15 所示。



图 1.14 向项目管理器中加入表



图 1.15 从项目管理器中移去或删除文件

#### 3. 在项目管理器中新建、修改文件

在项目管理器中新建和修改文件是非常容易的,只要选择所要创建的文件类型,然后单击【项目管理器】窗口右侧的【新建】或【修改】按钮即可。Visual FoxPro 6.0 将会打开合适的设计器,从而方便地完成设计任务。

#### 4. 为项目文件添加说明

创建或添加新的文件时,可以为文件添加描述性的说明文字。在选择文件时,这些说明信息将显示在【项目管理器】窗口的底部。

在【项目管理器】窗口中选择要添加说明的文件，选择【项目】菜单中的【编辑说明】命令，打开【说明】对话框，输入文字说明，单击【确定】按钮，如图 1.16 所示。

### 1.3.4 操作【项目管理器】窗口

#### 1. 改变、展开和折叠窗口

【项目管理器】窗口是一个独立的窗口，可根据需要移动【项目管理器】窗口的位置。拖动窗口边线或对角线可调整窗口大小，也可单击【项目管理器】窗口右上角的向上箭头，将窗口折叠起来。再次单击向下箭头又可以展开，如图 1.17 所示。



图 1.16 【说明】对话框



图 1.17 折叠【项目管理器】窗口

当【项目管理器】窗口折叠之后，可选择某一选项卡，将此选项卡从窗口中拖出，使其成为一个独立的浮动选项卡。如图 1.18 所示，当把【数据】、【文档】选项卡拖开后，该选项卡便可独立移动，此时【项目管理器】窗口中原来的【数据】、【文档】选项卡变为无效的灰色。浮动选项卡中“图钉”按钮的作用是：当按下“图钉”按钮就能使该选项卡始终显示在屏幕最上层。

#### 2. 项目信息

当创建新的项目后，系统会自动打开以该项目名命名的【项目管理器】窗口，系统同时也会自动在 Visual FoxPro 6.0 系统菜单中添加一个【项目】菜单。在【项目】菜单中选择【项目信息】命令，就可进入【项目信息】对话框，如图 1.19 所示。在【项目信息】对话框中选择相应的选项卡，便可浏览和编辑与项目相关的信息。



图 1.18 分离【项目管理器】窗口的选项卡

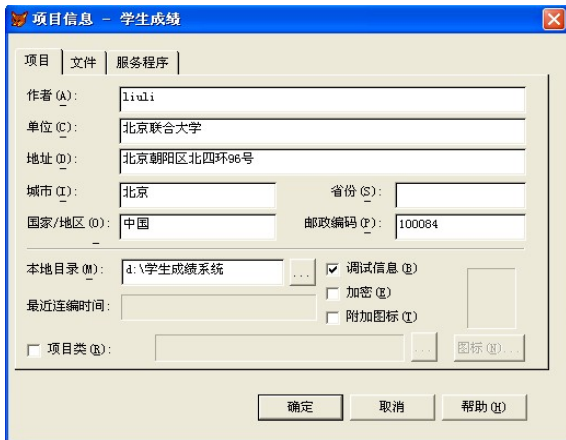


图 1.19 【项目信息】对话框

### 1.3.5 Visual FoxPro 6.0 的辅助设计工具

在 Visual FoxPro 应用软件中提供了很多设计工具，其中 Visual FoxPro 提供了 3 类支持可视化设计的辅助工具——设计器、向导、生成器，可以快速高效地开发一些应用程序。

#### 1. 设计器

Visual FoxPro 提供的一系列设计器，为用户提供了一个友好的图形界面操作环境，用以创建、定制、编辑数据库结构、表结构、报表格式、应用程序组件等。

利用项目管理器，可以快速访问 Visual FoxPro 的各种设计器。这些工具使得创建表、表单、数据库、查询和报表来管理数据变得轻而易举。

#### 2. 向导

向导是一种快捷设计工具。它通过一组对话框依次与用户对话，引导用户分步完成 Visual FoxPro 的某项任务，例如创建一个新表、建立一项查询或设置一个报表的格式等。

Visual FoxPro 有 20 余种向导工具。从创建表、视图、查询等数据文件，到建立报表、标签、图表、表单等 Visual FoxPro 文档，直至创建 Visual FoxPro 的应用程序、SQL 服务器上的数据库等操作，均可使用相应的向导工具来完成。例如，若需要创建一个新表，可先用表向导来创建，然后再用表设计器进行修改。

#### 3. 生成器

生成器是带有选项卡的对话框，用于简化对表单等复杂控件和参照完整性代码的创建与修改过程。在 Visual FoxPro 应用程序的构件中生成并加入某类控件，例如生成一个组合框或生成一个列表框等。每次向表单添加新控件，Visual FoxPro 都显示一个适当的生成器。

以上 3 类辅助工具全部使用图形交互界面。通过直观的人机交互操作，就可使用户轻松地完成应用程序的界面设计任务。不仅如此，所有上述工具的设计结果，都能够自动生成 Visual FoxPro 代码，使用户可以摆脱面向对象程序设计的繁琐编码任务，轻松地建立起自己的 Visual FoxPro 应用程序。

### 练习与操作

**练习 1** 在 D: 盘上建立一个文件夹，命名为“学生成绩系统”，即 D:\学生成绩系统。以后所有的练习内容都保存在这个文件夹中。

**练习 2** 创建“学生成绩”项目，具体步骤如下。

- (1) 选择菜单栏上的【文件】→【新建】命令，在弹出的【新建】对话框中选择文件类型为【项目】，同时单击【新建文件】按钮，如图 1.20 所示。
- (2) 单击【新建文件】按钮后，弹出保存文件的对话框，在这里选择保存文件的路径为 D: 盘的“学生成绩系统”文件夹，并在【项目文件】中输入“学生成绩”（默认值为“项目 1”），在【保存类型】中选择“项目 (\*.pjx)”。以上 3 个参数都设置完后，单击【保存】按钮。
- (3) 保存“学生成绩”项目后，弹出【项目管理器—学生成绩】窗口，如图 1.21 所示。

这样，就创建好了“学生成绩”项目。此时可以在 D: 盘的“学生成绩系统”文件夹看到新加了“学生成绩.pjt”（项目备注文件）和“学生成绩.pjx”（项目文件）两个文件。



图 1.20 【新建】对话框



图 1.21 【项目管理器—学生成绩】窗口

## 1.4 学生成绩管理系统案例分析

### 1.4.1 案例设计分析

学生成绩管理是学校教学管理部门的一项重要任务。学生成绩管理包括对学生信息、教师信息、课程及学习成绩等的管理，设计该系统的目的就是利用计算机的快速查询和运算功能，替代管理人员对数据的手工处理。通过后期各案例项目的操作，使读者初步掌握 Visual FoxPro 数据库应用技能。下面建立的学生成绩管理系统就是一个基于 Visual FoxPro 开发的小型数据库应用系统，能够实现对教师、学生基本信息和学习成绩的输入、查询、维护及输出等功能。

以下建立的学生成绩管理系统，包含了 6 个有相互关系的表，并设置了各个表中某些字段的属性。通过建立不同类型的视图，实现了对数据的浏览、统计和添加等操作。通过建立一个主菜单列出所要实现的各种功能，利用创建的不同类型的表单实现与用户进行交互操作的友好界面。通过建立报表，实现了对成绩等数据的分析、整理与输出。最后，通过建立主程序和项目的连编，形成一个可执行的学生成绩管理系统，达到了快速完成选课、查分、统计等操作的效果。

在开始程序设计前，应先将程序的总体结构以层次图的形式表示出来，便于对程序进行分层编程和实现，如图1.22所示。

图的第一层为系统层，通常对应主程序。第二层为子系统层，一般起划分系统功能的作用，通常对应主菜单。第三层为功能层，对应菜单项，一般通过表单的调用来实现某些特定的功能。

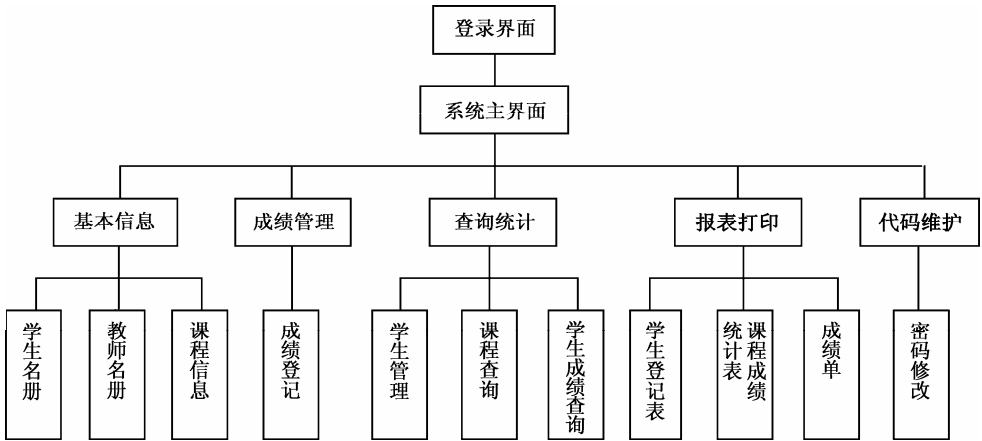


图 1.22 系统功能结构图

### 1.4.2 学生成绩管理系统的功能模块设计

#### 1. 表设计

- ① 学生表。字段包括：学号、姓名、专业编号、性别、出生日期、入学时间、入学成绩、团员否、照片、简历。学号为主索引（主关键字），为专业编号建立索引，通过该索引和专业表建立关联。
- ② 教师表。字段包括：教师编号、教师姓名、性别、职称、毕业学校、课程编号、工资、政府津贴。教师编号为主索引。
- ③ 课程表。字段包括：课程编号、课程名称、学时、学分、课程性质、备注。课程编号为主索引。
- ④ 学生选课表。字段包括：学号、课程编号、开课时间、成绩。学号、课程编号和开课时间共同做主索引，分别为学号、课程编号、开课时间、成绩建立索引，通过学号和学生表建立关联，通过课程编号和课程表建立关联，开课时间和成绩索引用于统计计算。
- ⑤ 教师任课表。字段包括：教师编号、课程编号。分别为教师编号和课程编号建立索引，通过教师编号和教师表建立关联，通过课程编号和课程表建立关联。
- ⑥ 专业表。字段包括：专业编号、专业名称、所属系、备注。为专业编号建立索引，通过专业编号和学生表建立关联。

#### 2. 表单设计

系统菜单下的大部分功能都是通过表单来实现的。除了系统的登录界面，表单还包括：学生专业.scx、教师情况表.scx、选课情况.scx、成绩登记.scx、按课程查询.scx、按学生查询.scx 及专业.scx 等表单。部分表单如图 1.23 和图 1.24 所示。

#### 3. 报表设计

主结构中的“报表打印”下有 3 个报表：学生登记表、成绩单、课程成绩统计表。报表浏览结果如图 1.25 和图 1.26 所示。

Form1

### 学生专业一览表

学号: 080301001    专业编号: 03    照片: 

姓名: 张跃林    专业名称: 数学

性别: ☒ 男 ☐ 女    入学成绩: 589

☒ 团员否    所属系: 基础

首页 上页 下页 末页 寻页 增页 删页 关闭

图 1.23 “学生专业”表单

Form1

### 成绩登记

学号: 080301001    姓名: 张跃林

课程编号: CC01    课程名称: VB.NET程序设计

开课时间: 02/20/2009    成绩: 96

首页 上页 下页 末页 寻页 增页 删页 关闭

图 1.24 “成绩登记”表单

报表设计器 - 成绩单.frx - 页面 1

### 成绩单

01/18/09

学号	姓名	性别	课程名称	考试成绩	平时成绩	总评成绩
070401032	刘慧琴	女	数据库原理及应用	92	80	89.6
070403128	赵晓飞	男	VB.NET程序设计	86	80	84.8
074202123	李林萍	女	大学计算机应用基础	90	80	88.0
080301001	张跃林	男	VFP数据库程序设计	76	80	76.8
080301001	张跃林	男	多媒体计算机技术	87	80	85.6
080301001	张跃林	男	VB.NET程序设计	96	80	92.8

图 1.25 “成绩单”报表

报表设计器 - 课程成绩统计表.frx - 页面 1

### 课程成绩统计表

01/18/09

课程编号: CC01    课程名称: VB.NET程序设计

学号	姓名	开课时间	成绩
080301001	张跃林	02/20/09	96

图 1.26 “课程成绩统计表”报表



4. 设置主文件

当用户运行应用程序时，Visual FoxPro 将启动主文件，然后主文件再依次调用所需要的应用程序的其他组件。一般最好的方法是为应用程序建立一个主程序。但是，使用一个表单作为主程序，可以将主程序的功能和初始界面集成在一起。

5. 基本表结构和记录

学生成绩管理系统的基本表结构如表1.7~表1.12所示，图1.27~图1.32 给出了“教师表”、“教师任课表”、“课程表”、“学生表”、“学生选课表”和“专业表”的结构示意图。

表 1.7 教师表

字段名	类型	宽度	小数位数	索引
教师编号	字符型	6		主索引
教师姓名	字符型	8		
性别	字符型	2		
职称	字符型	8		
毕业学校	字符型	16		
课程编号	字符型	4		
工资	数值型	8	2	
政府津贴	逻辑型	1		

表 1.8 教师任课表

字段名	类型	宽度	小数位数	索引
课程编号	字符型	4		普通索引
教师编号	字符型	6		普通索引

表 1.9 课程表

字段名	类型	宽度	小数位数	索引
课程编号	字符型	4		主索引
课程名称	字符型	20		
学时	数值型	3	0	
学分	数值型	2	0	
课程性质	字符型	8		
备注	备注型	4		

表 1.10 学生表

字段名	类型	宽度	小数位数	索引
学号	字符型	12		主索引
姓名	字符型	8		



(续表)

字段名	类型	宽度	小数位数	索引
专业编号	字符型	2		
性别	字符型	2		
出生日期	日期型	8		
入学时间	日期型	8		
入学成绩	数值型	3	0	
团员否	逻辑型	1		
照片	字符型	4		
简历	备注型	4		

表 1.11 学生选课表

字段名	类型	宽度	小数位数	索引
学号	字符型	12		普通索引
课程编号	字符型	4		普通索引
开课时间	日期型	8		普通索引
成绩	数值型	3	0	普通索引

教师表							
教师编号	教师姓名	性别	职称	毕业学校	课程编号	工资	政府津贴
JC0101	陈得江	男	教授	北京大学	CC01	3310.10	T
JC0202	赵惠兰	女	副教授	清华大学	CC01	2587.00	F
JC0303	刘大林	男	讲师	上海交通大学	CC02	2000.00	F
JS0101	秦健中	男	副教授	浙江大学	CS01	3120.00	T
JS0202	吴莲敏	女	副教授	复旦大学	CS01	1500.00	F
CC0101	赵晓兰	女	教授	同济大学	CS02	3334.00	F
CC0202	李月敏	女	讲师	吉林大学	CS02	1678.00	F
CC0303	王克强	男	讲师	北京理工大学	CC03	1584.00	F
CC0404	刘雪燕	女	副教授	北京交通大学	CS01	2345.00	F
CS0101	林星明	男	教授	北京林业大学	CS02	3254.00	F
CS0202	赵华林	男	副教授	厦门大学	CS02	2786.00	F
CS0303	季严冬	男	教授	哈尔滨工业大学	CC03	3786.00	T

图 1.27 “教师表”记录

教师任课表	
课程编号	教师编号
CC01	JC0101
CC01	JC0202
CC02	JC0101
CC02	JC0303
CC03	CC0303
CC03	JC0303
CS01	JS0101
CS01	JS0202
CS02	JS0101
CS02	CC0101
CS02	CC0202
CS01	CC0404
CS02	CS0101
CS02	CS0202

图 1.28 “教师任课表”记录

课程表					
课程编号	课程名称	学时	学分	课程性质	备注
CC01	VB.NET程序设计	60	4	必修课	memo
CC02	数据库原理与应用	60	4	必修课	memo
CC03	多媒体计算机技术	45	3	必修课	memo
CS01	VFP数据库程序设计	75	5	必修课	memo
CS02	大学计算机应用基础	30	2	指定选修	memo

图 1.29 “课程表”记录

学生表										
学号	姓名	专业编号	性别	出生日期	入学时间	入学成绩	团员否	照片	简历	
080301001	张跃林	03	男	10/23/83	09/01/08	589	T	Gen	Memo	
080301020	张文斌	03	男	05/20/84	09/01/08	593	T	Gen	memo	
080302045	陈江城	03	男	08/04/83	09/01/08	598	T	Gen	Memo	
070403128	赵晓飞	04	男	04/06/83	09/01/07	521	F	Gen	memo	
080401081	孟江浩	04	男	09/20/84	09/01/08	623	T	Gen	memo	
070401032	刘慧琴	04	女	07/30/83	09/01/07	599	T	Gen	memo	
084104112	郝明雅	41	女	10/31/83	09/01/08	471	F	Gen	memo	
074101067	李巧兰	41	女	05/03/83	09/01/07	473	F	Gen	memo	
084201002	夏利华	42	女	09/04/83	09/01/08	497	F	Gen	memo	
074202123	李林萍	42	女	03/05/84	09/01/07	516	T	Gen	memo	
084205255	李雅迪	42	女	07/08/83	09/01/08	478	T	Gen	memo	
084202152	殷 俐	42	女	01/02/83	09/01/08	506	T	Gen	memo	

图 1.30 “学生表”记录

学生选课表				
学号	课程编号	开课时间	成绩	
080301001	CC01	02/20/09	96	
080301001	CC03	02/20/09	87	
080301001	CS01	02/23/08	76	
080301020	CC02	02/20/09	87	
080301020	CS02	02/20/09	88	
084201002	CC02	02/20/09	55	
084201002	CC03	02/20/09	89	
074202123	CS02	09/01/09	90	
070403128	CC01	02/23/08	86	
084104112	CS02	02/23/08	50	
084104112	CS02	02/20/09	67	
084205255	CC02	02/23/08	91	
084202152	CC03	02/23/08	85	
084202152	CS01	02/23/08	78	
084202152	CS02	09/03/08	75	

图 1.31 “学生选课表”记录

表 1.12 专业表

字段名	类型	宽度	小数位	索引
专业编号	字符型	4		主索引
专业名称	字符型	12		
所属系	字符型	12		
备注	备注型	4		

专业表				
专业编号	专业名称	所属系	备注	
03	水动	水利	Memo	
04	农田	水利	Memo	
05	水资源	水利	memo	
41	自动化	电力	memo	
42	热动	电力	memo	
01	计算机应用	计算机	memo	

图 1.32 “专业表”记录

6. 案例成果展示

要求将本案例设计保存在“学生成绩系统”文件夹中，其中包括一个“学生.dbc”数据库文件；6 个数据库表，即“教师表.dbf”、“教师任课表.dbf”、“课程表.dbf”、“学生表.dbf”、“学生选课表.dbf”和“专业表.dbf”；以及表单、报表、主菜单和主程序等。

双击“学生成绩系统.app”应用程序文件，进入登录界面，单击【进入系统】按钮后，进入系统主界面。学生成绩管理系统登录界面和系统主界面如图 1.33 和图 1.34 所示。



图 1.33 学生成绩管理系统登录界面



图 1.34 学生成绩管理系统的主界面

### 1.4.3 学生成绩管理系统的设计要求

Visual FoxPro 数据库程序设计是一门实践性很强的课程，学生在学习这门课程时，要十分重视实验环节。我们将“学生成绩管理系统”作为实验主线，将“学生成绩管理系统”中的各知识点分解到各个实验点中。通过各个实验的操作和设计，最终完成“学生成绩管理系统”的软件开发。

要求学生独立上机操作，独立编写程序、调试程序，并独立完成该小型系统程序的开发与设计。学生要在每一个实验完成后保存数据、撰写实验报告，学期末撰写“学生成绩管理系统”开发报告。

## 思考与练习 1

### 一、思考题

1. 数据与信息的异同是什么？
2. 数据库系统与文件系统的主要区别是什么？
3. 层次模型、网状模型和关系模型的区别是什么？
4. 【项目管理器】窗口由哪几部分组成？
5. 以“学生”为文件名建立一个项目文件的步骤是什么？

### 二、选择题

1. Visual FoxPro 关系数据库管理系统能够实现的三种基本关系运算是\_\_\_\_。  
A. 选择、投影、连接  
B. 索引、排序、查找  
C. 选择、索引、联系  
D. 差、交、并
2. Visual FoxPro 是一种关系数据库管理系统，所谓关系是指\_\_\_\_。  
A. 数据模型符合满足一定条件的二维表格式  
B. 表中的各个记录之间有联系

- C. 表中的各个字段之间有联系  
D. 数据库中的一个表与另一个表有联系
3. 在 Visual FoxPro 中, 数据库完整性一般包括\_\_\_\_\_。
- A. 实体完整性、域完整性  
B. 实体完整性、域完整性、参照完整性  
C. 实体完整性、域完整性、数据库完整性  
D. 实体完整性、域完整性、数据表完整性
4. 数据库 (DB)、数据库系统 (DBS)、数据库管理系统 (DBMS) 之间的关系是\_\_\_\_\_。
- A. DB 包括 DBS 和 DBMS  
B. DBS 包括 DB 和 DBMS  
C. DBMS 包括 DBS 和 DB  
D. 三者等级, 没有包含关系
5. 二维表中的列称为关系的\_\_\_\_\_; 二维表中的行称为关系的\_\_\_\_\_。
- A. 元组, 属性  
B. 列, 行  
C. 行, 列  
D. 属性, 元组
6. 选择 Visual FoxPro 系统的【工具】主菜单, 在【选项】对话框的【文件位置】选项卡中可以设置\_\_\_\_\_。
- A. 菜单生成器文件的位置  
B. 程序代码的颜色  
C. 日期和时间的显示方式  
D. 显示快捷菜单命令的对话框
7. 在【项目管理器】窗口中, 如果某个文件前面出现加号标志, 表示\_\_\_\_\_。
- A. 该文件中只有一个数据项  
B. 该文件中有一个或多个数据项  
C. 该文件中有多个数据项  
D. 该文件不可用
8. 表单文件的扩展名为\_\_\_\_\_。
- A. .scx  
B. .prg  
C. .hlp  
D. .idx
9. 下列启动 Visual FoxPro 向导的方法中, 正确的是\_\_\_\_\_。
- A. 选择【工具】菜单的【向导】子菜单中的命令  
B. 通过【新建】对话框  
C. 单击工具栏上的“向导”按钮  
D. 以上方法均正确
10. 扩展名为“.prg”的程序文件可在【项目管理器】窗口的\_\_\_\_\_选项卡中进行显示和管理。
- A. 数据  
B. 文档  
C. 代码  
D. 其他

### 三、填空题

1. 数据库是存储在计算机存储设备上的、结构化的相关数据集合。它不仅包括\_\_\_\_\_, 而且还包括\_\_\_\_\_。
2. 关系数据库中每个关系的形式是\_\_\_\_\_。
3. 数据库系统的核心是\_\_\_\_\_。
4. 如果表中的一个字段不是本表的\_\_\_\_\_或\_\_\_\_\_, 而是另外一个表的\_\_\_\_\_或\_\_\_\_\_, 这个字段(属性)就称为外部关键字。

5. 任何一个数据库管理系统都是基于\_\_\_\_\_建立的。数据库管理系统支持的数据模型分三种：\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_。
6. 通过 Visual FoxPro 窗口标题栏左端的控制按钮，可以\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_。
7. 安装好 Visual FoxPro 后，系统提供了一个默认的工作环境，如果要定义自己的工作环境，应选择\_\_\_\_\_菜单中的\_\_\_\_\_命令。
8. 在项目管理器中也可以将应用系统编译成一个扩展名为\_\_\_\_\_的应用文件或\_\_\_\_\_的可执行文件。
9. 【项目管理器】窗口共有\_\_\_\_\_个选项卡，依次为\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_；其中，\_\_\_\_\_选项卡中包括“数据库”、“自由表”和“查询”三项。
10. 【项目管理器】窗口中的【移去】按钮有两个功能：一是\_\_\_\_\_；二是\_\_\_\_\_。

# 第 2 章 数据与数据运算

计算机程序是用来描述一些数据及作用在这些数据上的操作。使用计算机解题，就是要用特定的程序设计语言可以描述的数据来模拟处理对象，并在其上施加该程序设计语言可以描述的操作。为了顺利编写程序代码，学习数据类型、常量、变量、表达式和常用函数等编程语言的基础知识是必要的。

## 2.1 数据与运算

### 2.1.1 数据类型

与其他程序设计语言类似，Visual FoxPro 也提供了丰富的数据类型，可以把数据存入各种类型的数据表、数组和内存变量中。Visual FoxPro 的数据类型分为两大类，一类适用于变量和数组；另一类适用于数据表中的字段。Visual FoxPro 定义了 13 种字段类型和 7 种数据类型。13 种字段类型是：字符型、数值型、浮点型、双精度型、整型、货币型、日期型、日期时间型、逻辑型、备注型、通用型、二进制字符型和二进制备注型；7 种数据类型是：字符型、数值型、货币型、日期型、日期时间型、逻辑型和通用型。

### 2.1.2 常量和变量

#### 1. 常量

常量是在命令或程序操作过程中可以直接引用，并在整个操作过程中其值保持不变的数项。在 Visual FoxPro 中常量的数据类型有 6 种，分别是：字符型、数值型、货币型、日期型、日期时间型、逻辑型。Visual FoxPro 中常量的数据类型和特征如表 2.1 所示。

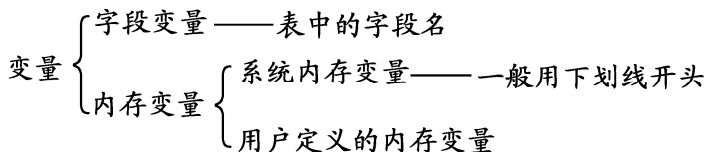
表 2.1 常量的数据类型和特征

常量	特征
字符型	用定界符括起来的由字符、空格或数字所组成的字符串（定界符有 3 种：" "、' '、[ ]）
数值型	又称为常量，可以是整数或实数（如 23、-16.8、1.37E+4、2.89E-3 等）
货币型	在数字前加上货币符号“\$”就是货币型常量（如\$254.56，表示货币值 254.56）
日期型	用花括号“{ }”作为定界符括起来（如{^2008/03/18}或{^2008-03-18}）
日期时间型	用花括号“{ }”作为定界符括起来（如{^2008/04/19 10:04am}、{^2008-03-21 11:32:14pm}）
逻辑型	用.T.、.t.、.Y.、.y. 表示逻辑值“真”，用.F.、.f.、.N.、.n.表示逻辑值“假”

#### 2. 变量

在命令操作和程序运行过程中其值允许变化的量称为变量，Visual FoxPro 中使用的变量分为字段变量、内存变量两种。每个变量都必须有一个变量名，变量名由字母、汉字、数字和下划线组成，但必须以字母或汉字开头。

### 1) 变量类型



① 字段变量。字段变量是数据表结构中的任意一个数据项。在一个数据表中，一个字段就是一个字段变量。字段变量在建立表结构时定义，在修改表结构时可重新定义，也可增加或删除字段变量。字段变量的类型有 13 种。字段变量是一种多值变量，假设一个数据表中有 10 条记录，那么该数据表的每一个字段就有 10 个取值，当使用字段变量时，它的当前值随着记录指针的移动而不断变化，以后章节中还要详细地对其进行讲解。

② 内存变量。内存变量是表结构之外独立存在于内存中的变量。内存变量可用来存储数据，定义内存变量时需要为它取名并赋初值，内存变量建立后存储于内存中。Visual FoxPro 中共定义了 6 种类型的内存变量：字符型、数值型、日期型、日期时间型、逻辑型和货币型。可分为简单变量和数组变量。

### 2) 内存变量赋值命令

内存变量赋值命令有两种格式：

- <内存变量> = <表达式>
- STORE <表达式> TO <内存变量表>

其功能是计算<表达式>，然后将计算结果赋给内存变量。

例如，定义内存变量 Z，其值为 abcde；再定义内存变量 z1、z2、z3 的值，均为 5\*7。使用下面两条命令定义变量：

```
Z="abcde"  && 将字符串 abcde 赋给变量 Z，结果 Z 值为 abcde，并成为字符型变量
STORE 5*7 TO z1, z2, z3  &&计算 5*7 得 35，结果 3 个变量值都是 35，都成为数值型变量
```

需要说明的是

- ① 内存变量的类型与所赋值的常量的类型相同。
- ② STORE 命令可包括多个内存变量，但变量与变量之间须用逗号分隔。该命令可将同值常量赋值给多个变量，而“=”命令仅可为一个内存变量赋值。
- ③ 符号“&&”是本命令行的注释，它与命令执行无关，常用来解释命令功能。

### 3) 表达式值显示命令

表达式值显示命令的格式如下所示：

```
?/?? <表达式表>
```

其功能是将表达式的值显示在屏幕上。需要说明的是

① 命令格式中的“?/??”可任选一个，它们的输出格式不同。“?”表示从屏幕下一行的第一列起显示结果，“??”表示从当前行的当前列起显示结果。例如：

```
? Z           && 在 Visual FoxPro 主窗口中显示 abcde
? z1          && 换一行后显示 35
?? "数据库应用" && 接着上一个命令显示结果"35"的后面显示"数据库应用"
```

② <表达式表>是表示用逗号隔开的多个表达式，命令执行时遇逗号就空一格。例如：

```
? "Z=",Z           && 显示"Z= abcde"
```

4) 数组变量

数组是具有相同名字、不同下标值的一组变量。数组中的每个数据值称为数组元素，每个数组元素在数组中的位置是固定的，可以通过称为下标的编号进行访问。数组在使用时要由 DIMENSION 或 DECLARE 命令来定义。这两个命令完全相同，都是用于建立一维或二维数组。在 Visual FoxPro 中，同一个数组元素在不同时刻可以存储不同类型的数据，在同一数组中，每个数组元素可以被赋予不同数据类型的数据。

数组变量的定义如下所示：

```
DIME Z(3)           && 建立包含 3 个元素的一维数组 Z
DECL DZ(9), DB(2, 3) && 建立一维数组 DZ 和二维数组 DB
```

定义数组后，就可以用 STORE 命令或赋值语句为数组的每个元素赋值了。例如：

```
DIME Z(5)           && 建立包含 5 个元素的一维数组 Z
Z(1)="XYZ"          && 给 Z(1)赋值，字符型，值：XYZ
Z(2)=45.34           && 给 Z(2)赋值，数值型，值：45.34
Z(3)={^2009-02-02}   && 给 Z(3)赋值，日期型，值：2009 年 2 月 2 日
Z(4)={^2009-02-02 9:00am} && 给 Z(4)赋值，日期时间型，值：2009 年 2 月 2 日上午 9 点
Z(5)=.F.             && 给 Z(5)赋值，逻辑型，值：假
```

2.1.3 运算符及表达式

运算符是表示数据之间运算方式的运算符号，一般根据处理的数据类型不同可分为算术运算符、字符运算符、关系运算符、逻辑运算符 4 种。表达式是由常量、变量、函数、运算符及圆括号组成的算式。表达式中的操作对象必须具有相同的数据类型，如果表达式中有不同类型的操作对象，则必须将它们转换成同种数据类型。

1. 算术表达式

算术表达式是由数值型变量、常量、函数和算术运算符组成的，用于对数值型数据进行常规的算术运算，如表 2.2 所示。

表 2.2 算术运算符

运算符	含义	优先级
()	括号	高 ↓ 低
**或^	乘方	
*, /	乘、除	
%	取模（或取余），取两数相除的余数	
+, -	加、减	

算术表达式如下所示：

```
X^4+6*X^2
SQRT(38.23)/INT(67.24)
32%4-X^2
```



2. 字符表达式

字符表达式是由字符型变量、常量、函数和字符运算符组成的，用于字符串的连接或者比较，如表 2.3 所示。

表 2.3 字符运算符

运算符	含义	说明
+	完全连接	连接两个字符串
-	不完全连接	连接两个字符串，并将第一个字符串尾部的空格移到第二个字符串的尾部
\$	字符串比较	判断第一个字符串是否是第二个字符串的子字符串

字符表达式如下所示：

```
LEN1="Visual FoxPro6.0"
LEN2="数据库教程"
LEN3="is"
? LEN1+LEN2      && 结果为: Visual FoxPro6.0 数据库教程
? LEN1-LEN2      && 结果为: Visual FoxPro6.0 数据库教程
? len(LEN1-LEN2) && 结果为: 28
? LEN3 $ LEN1    && 结果为: .T.
```

3. 日期时间表达式

日期时间表达式是由日期时间型变量、常量、函数和日期时间运算符组成的。  
日期时间运算符有

- “+”：添加一个天数或秒数
- “-”：减少一个天数或秒数

日期时间表达式如下所示：

```
? {^2009-04-04}+10      && 结果为: 04/14/09
? {^2009-04-30}-15      && 结果为: 04/15/09
? {^2009-08-04 10:10a}+10  && 结果为: 08/04/09 10:10:10 AM
? {^2009-08-04 10:35p}-10  && 结果为: 08/04/09 10:34:50 PM
```

4. 关系表达式

关系表达式用于数值、字符、日期型数据的比较运算。关系表达式的运算优先级相同，如表 2.4 所示。

表 2.4 关系运算符

关系运算符	含义	关系运算符	含义
<	小于	<=	小于等于
>	大于	>=	大于等于
=	等于	#, <>, !=	不等于
==	精确相等		

关系表达式如下所示：

```
? "abc"<>"ABC"
? "abcde"="abcd"
? "abcd"="abcde"
"abcd"=="abcd"
```

```
&& 值为.T.
&& 值为.T.
&& 值为.F.
&& 值为.T.
```

5. 逻辑表达式

逻辑表达式是由逻辑型变量、常量、函数和字符运算符组成的，用来对逻辑型数据进行各种逻辑运算，形成各种简单的逻辑结果，如表 2.5 所示。

表 2.5 逻辑运算符

运算符	含义	优先级
( )	分组符号	高 ↑ 低
.NOT.	逻辑非	
.AND.	逻辑与	
.OR.	逻辑或	

逻辑表达式如下所示：

```
? 18>26 .AND. 31>17
? 18/2>7 .OR. "abc"<>"ABC" .AND. NOT.F.
```

```
&& 值为.F.
&& 值为.T.
```

6. 空值

空值（null）是一个重要的概念。空值就是没有任何值。对于数值，它非零；对于字符，它非空格串；对于逻辑，它非真非假。在应用中，空值的概念是十分有意义的。例如，当不知道年龄时，不能填零；当不知道姓名时，也不能填空字符串；等等。空值可表示为：.NULL.。

内存变量、数组变量、字段变量均可以赋以空值。变量赋以空值后，其类型不变。也就是说空值不是一个数据类型。例如：

```
X=" "
Y=" "
Z=.NULL.
```

```
&& X 为空格字符串
&& Y 为空字符串
&& Z 赋以空值，但不改变其类型
```

再例如：

```
a=.NULL.
? a+4
b=.t.
? b and .NULL.
? b or .NULL.
? not .NULL.
```

```
&& 结果为: .NULL.
&& 结果为: .NULL.
&& 结果为: .T.
&& 结果为: .NULL.
```

练习与操作

练习 1 在命令窗口中依次执行以下命令，并观察运行结果。

```
? -100.2
```

```
? "25fgh%&$"
? {^2009-03-03}
```

练习2 在命令窗口中依次执行以下命令，并观察运行结果。

```
x1=8*4
x2="pqr"
xy=.t.
xz={^2009-03-01}
? x1,x2,xy,xz
? "ad" $ "abcd" or (1.1+2)^3>66
```

练习3 定义一个一维数组 A(5)，分别给数组的每个元素赋值为

```
123. 54 "123.54" {^2009-03-09 10:12:34am} .T. .NULL.
```

执行以下命令，并观察运行结果。

```
? A(1)A(2),A(3),A(4),A(5)
```

练习4 依次执行以下命令，并观察运行结果。

```
? 3>4
? "abcd">"ad"
set exact off
? "abcdef"="abc"
? "abcdef"=="abd"
set exact on
? "abcdef"="abc"
? "abcdef"=="abc"
```

练习5 依次执行以下命令，并观察运行结果。

```
? not (5>6)
? not (5>6) and "abc" > "ad" or $100<$200
```

**提示：**Visual FoxPro 使用 set exact on/off 命令打开或关闭精确比较。当设为 off 时，比较是从两个字符串最左边开始的，然后逐个比较字符，一直到较短字符串结束，若进行比较的等号右边的字符串是左方字符串的子串时，认为两个字符串相等，否则不相等。如果设为 on，进行比较的字符串只有长度相等且逐个字符相同时，才认为两字符相等。系统默认设置为 off。

## 2.2 常用函数

Visual FoxPro 系统为用户提供了十分丰富的函数。灵活运用这些函数，不仅可以简化许多运算，而且能够加强和完善 Visual FoxPro 的许多功能。Visual FoxPro 提供了许多不同用途的标准函数，以帮助用户完成各种工作。函数的一般格式如下所示：

函数名(自变量表)

函数与表达式类似，也是一种运算。只不过我们不必写出具体怎样运算，而是由每个函数来处理其运算方法。我们只需写上函数名，在随后的圆括号内写上自变量，就可以根据自变量的值，由该函数得到一个结果值，称之为函数值。每个函数的自变量个数、类型、

书写顺序都必须按照相关的规定。Visual FoxPro 提供了几百种函数，下面我们只介绍最常用的一些函数。

## 2.2.1 数学运算函数

### 1. 平方根函数SQRT()

格式: SQRT(<数值表达式>)

功能: 计算并返回<数值表达式>的算术平方根。

示例: ? SQRT(5\*5)

5.00

### 2. 绝对值函数ABS()

格式: ABS(<数值表达式>)

功能: 计算并返回<数值表达式>的绝对值。

示例: ? ABS(-213.27)

213.27

### 3. 四舍五入函数ROUND()

格式: ROUND(<数值表达式>, <有效位数>)

功能: 对<数值表达式>的值按指定的<有效位数>进行四舍五入。

示例: ? ROUND(2.34259, 4), ROUND(5234.5678, 0), ROUND(5234.567, -1)

2.3426      5235      5230

### 4. 取整函数INT()

格式: INT(<数值表达式>)

功能: 计算并返回<数值表达式>的整数部分。

示例: ? INT(9.987); INT(-37.93)

9      -37

### 5. 求模函数MOD()

格式: MOD(<被除数>, <除数>)

功能: 返回<被除数>除以<除数>得到的余数值。

说明: 在求模运算中应注意以下几点:

- 除数不能为 0。
- 除数为正数, 返回正数; 如果为负数, 返回负数。
- 如果被除数与除数能够整除, 结果为 0。
- 如果被除数与除数不能整除, 且被除数与除数同号, 则结果为被除数除以除数而得到的余数, 即  $\text{MOD}(X1, X2) = X1 - \text{INT}(X1 / X2) * X2$ 。
- 如果被除数与除数不能整除, 且被除数与除数异号, 则结果为被除数除以除数而得到的余数再加上除数, 即  $\text{MOD}(X1, X2) = X1 - \text{INT}(X1 / X2) * X2 + X2$ 。

示例: ? MOD(13,3)	&& 结果为: 1
? MOD(-13,-3)	&& 结果为: -1
? MOD(-13,3)	&& 结果为: 2
? MOD(13,-3)	&& 结果为: -2

## 2.2.2 字符处理函数

### 1. 取子串函数SUBSTR( )

格式: SUBSTR(<字符串表达式>, <起始位置> [, <长度>])

功能: 从<字符串表达式>中的<起始位置>截取子串, <长度>为所截取的子串的长度。

示例: CN="数据库程序设计"

```
? SUBSTR(CN,5,2),SUBS(CN,1,4),SUBS(CN,13,3)
库 数据 计
```

若省略<长度>, 或者<长度>超过从<起始位置>到末尾的长度, 则截取的子串为从<起始位置>到<字符串表达式>末尾的所有字符。例如:

```
? SUBSTR("VFP 程序设计教程", 8)
设计教程
```

### 2. 求字符串长度函数LEN( )

格式: LEN(<字符串表达式>)

功能: 返回<字符串表达式>中所包含的字符个数, 即字符串长度。

示例: ? LEN("Visual FoxPro6.0")  
16

### 3. 搜索子串位置函数AT( )

格式: AT(<字符串 1>, <字符串 2> [, <数值表达式>])

功能: 寻找<字符串 1>在<字符串 2>中首次出现的起始位置。若<字符串 2>中不包含<字符串 1>, 则返回值为零。若有<数值表达式> (假设数值表达式为 K), 则寻找<字符串 1>在<字符串 2>中第 K 次出现的位置。

示例: X="Visual FoxPro 6.0"

? AT("Fox",X)	&& 显示结果为 8
? AT("fox",X)	&& 显示结果为 0
? AT("o",X)	&& 显示结果为 9
? AT("o",X,2)	&& 显示结果为 13

### 4. 空格函数SPACE( )

格式: SPACE(<数值表达式>)

功能: 产生与<数值表达式>的值相同的空格数。

示例: ? "程序"+SPACE(4)+"设计"  
程序 设计      && 在"程序"和"设计"之间产生了 4 个空格字符

5. 删除空格函数 (LTRIM( )、RTRIM( )、ALLTRIM( ))

1) 删除前导空格: LTRIM( )

格式: LTRIM(字符串)

功能: 删除字符串前的所有空格。

示例: ? LTRIM(" 程序设计 ")

程序设计                                      && " 程序设计 "前的 4 个空格被删除掉了

2) 删除后导空格: RTRIM( )

格式: RTRIM(字符串)

功能: 删除字符串后的所有空格。

示例: ? RTRIM(" 程序设计 ")

程序设计                                      && " 程序设计 "后的 4 个空格被删除掉了

3) 删除前后空格: ALLTRIM( )

格式: ALLTRIM(字符串)

功能: 删除字符串前后的所有空格。

示例: ? ALLTRIM(" 程序设计 ")

程序设计                                      && " 程序设计 "前后的所有空格都被删除掉了

6. 宏代换函数&

格式: &<字符型内存变量>

功能: 取<字符型内存变量>的值。

说明:

- 宏代换是一种间接取值的操作, 在&符号后面必须紧跟(无空格)一个已被赋过值的字符型内存变量的名字。
- 若&<字符型内存变量>与后面的字符之间无空格分界, 应加上“.”符号作为分界符。
- 宏代换中可以嵌套另一个宏代换, 但不能嵌套自己。例如, X=“&X”的写法是错误的。
- 对于数字字符串, 可以通过&函数使其与其他数字进行计算。

示例: 求宏代换。

```
<1> CH="X"
    X=15
    ? &CH                                && 结果为: 15
<2> TT="学生成绩表"
    USE  &TT                             && 结果为: 打开了"学生成绩表.dbf"表文件
<3> X="2008"
    M="+"
    Y="&X.&M.1"
    ? Y                                    && 结果为: 2008+1
    ? &Y                                  && 结果为: 2009
```

## 2.2.3 日期和时间函数

### 1. 系统日期函数DATE()

格式: DATE()

功能: 返回当前系统日期。

示例: 设当前系统日期为 2009/04/23, 则有

```
? DATE()  
04/23/09
```

### 2. 系统时间函数TIME()

格式: TIME()

功能: 返回当前系统时间。

示例: 设当前系统时间为上午 11 点 30 分 25 秒, 则有

```
? TIME()  
11:30:25
```

### 3. 系统日期时间函数DATETIME()

格式: DATETIME()

功能: 返回系统当前的日期和时间。

示例: 同上例的日期和时间, 则有

```
? DATETIME()  
04/23/09 11:30:25 AM
```

### 4. 求年函数YEAR()

格式: YEAR()

功能: 返回系统当前的年份。

示例: 设当前系统日期为 2009/04/23, 则有

```
? YEAR(DATE())  
2009
```

## 2.2.4 测试函数

### 1. 测试文件头函数BOF()

格式: BOF()

功能: 该函数用于测试指定文件中的记录指针是否指向文件头。如果是则返回真值, 否则返回假值。

示例: 测试文件记录是否指向文件头。

```
USE 学生表.dbf
? BOF()      && 显示结果为: .F.
SKIP -1      && 移动记录指针到上一条记录命令
? BOF()      && 显示结果为: .T.
```

## 2. 测试文件尾函数EOF()

格式: EOF()

功能: 该函数用于测试指定文件中的记录指针是否指向文件尾。如果是则返回真值, 否则返回假值。

示例: 测试文件记录是否指向文件尾。

```
USE 学生表.dbf
GOTO BOTTOM
? EOF( )      && 显示结果为: .F.
SKIP          && 移动记录指针到下一条记录命令
? EOF( )      && 显示结果为: .T.
```

## 3. 测试当前记录号函数RECNO()

格式: RECNO()

功能: 测试指定表文件中的记录指针指向的记录号。若指定的工作区无打开的表文件, 函数返回值为 0; 若指定的表文件无记录或记录指针位于文件头, 函数返回值为 1; 若记录指针指向文件尾, 函数返回值为末记录号加 1。

示例: 测试记录指针的当前位置 (假设学生档案表中有 10 条记录)。

```
USE 学生表.dbf
GOTO BOTTOM
? RECNO( )      && 显示结果为: 10
SKIP
? RECNO( )      && 显示结果为: 11
GO TOP
? RECNO( )      && 显示结果为: 1
```

## 4. 数据类型测试函数VARTYPE()

格式: VARTYPE(<表达式>[, <逻辑表达式>])

功能: 测试<表达式>的数据类型, 返回用字母代表的数据类型。函数值为字符型。未定义或错误的表达式返回字母 U。若表达式是一个数组, 则根据第一个数组元素的类型返回字符串。

示例: 测试记录指针的当前位置 (假设学生档案表中有 10 条记录)。

```
a=DATE( )
b=NULL
? VARTYPE($385), VARTYPE("FoxPro"), VARTYPE(a, .T. ), VARTYPE(b)
&& 输出为: Y C D X
```

提示: 各种数据类型的代表符号为

C—字符型或备注型 N—数值型、整型、浮点型或双精度型 Y—货币型  
L—逻辑型 O—对象 G—通用型 D—日期型 T—日期时间型 X—空值 U—未知



## 5. 判断值介于两个值之间的函数BETWEEN( )

格式: BETWEEN (<被测表达式>, <下限表达式>, <上限表达式>)

功能: 判断表达式的值是否介于相同数据类型的两个表达式的值之间。BETWEEN( )首先计算表达式的值。如果一个字符、数值、日期、表达式的值介于两个相同类型表达式的值之间, 即<被测表达式>的值大于或等于<下限表达式>的值、小于或者等于<上限表达式>的值, BETWEEN( )将返回“.T.”值, 否则返回“.F.”。

示例: Gz=375

? BETWEEN(Gz, 260, 650)

&& 显示结果为: .T.

## 6. 条件函数IIF( )

格式: IIF (<逻辑型表达式>, <表达式 1>, <表达式 2>)

功能: 若<逻辑型表达式>的值为.T., 函数值为<表达式 1>的值, 否则为<表达式 2>的值。

示例: xb ="女"

? IIF(xb="男", 1, IIF(xb="女", 2, 3)) && 显示结果为: 2

## 2.2.5 转换函数

Visual FoxPro 提供了多种转换函数, 各类转换函数如图 2.1 所示。

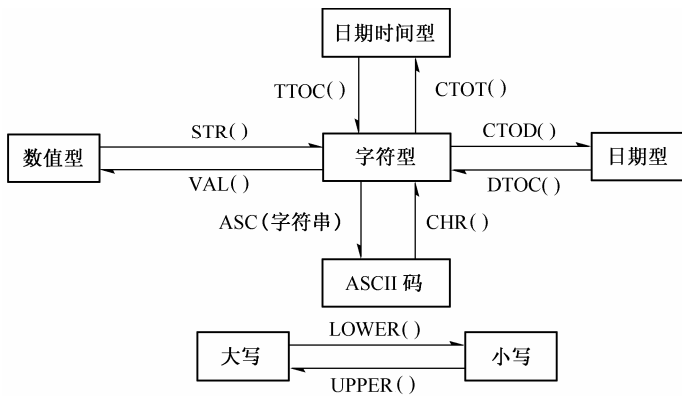


图 2.1 各类转换函数

### 1. 数值与字符串互换函数 (STR( )、VAL( ))

#### 1) 数值转换为字符串函数STR( )

格式: STR (<数值表达式> [, <长度> [, <小数位数> ]])

功能: 将<数值表达式>的值转换为由<长度>和<小数位数>指定的字符串。如果<长度>值比<数值表达式>的整数位数还要小, 则返回由<长度>个“\*”组成的字符串。

示例: ? STR(1324.46, 6, 1)

1324.5

? STR(1324.46, 3, 1), STR(1324.46, 8, 3)

\*\*\* 1324.460

## 2) 字符串转换为数值函数VAL()

格式: VAL(<字符串表达式>)

功能: 将<字符串表达式>转换为数值型数据。

示例: ? VAL("432"+"18")

43218.00

? VAL("512GH"), VAL("AB804"), VAL("ASD")

512.00    0.00    0.00

## 2. 字符型与日期型互换函数 (CTOD()、DTOC())

### 1) 字符型转换为日期型函数CTOD()

格式: CTOD(<字符串表达式>)

功能: 将“月/日/年”格式的<字符串表达式>转换为日期型数据。

示例: DA="02/12/09"

? CTOD(DA)

02/12/09

### 2) 日期型转换为字符型函数DTOC()

格式: DTOC(<日期>)

功能: 将<日期>转换为相应的字符串。

示例: D={^2009/03/24}

DT=DTOC(D)

? DT

?? LEN(DT)

03/24/09        8

## 3. 字符型与ASCII码互换函数 (ASC()、CHR())

### 1) 字符型转换为ASCII码函数ASC()

格式: ASC(<字符串表达式>)

功能: 将<字符串表达式>的第1个字符转换为ASCII码数据。

示例: ? ASC("abcdef")

97

### 2) ASCII码转换为字符型函数CHR()

格式: CHR(数值)

功能: 将ASCII码数值转换为相应的字符。

示例: ? CHR(67)

C

## 2.2.6 用户自定义对话框MESSAGEBOX()函数的使用

MESSAGEBOX()函数用于显示短信息。MESSAGEBOX()函数显示的小窗口不具有什么功能,实际上将其认为是一个对话框。

对话框是用户与应用程序之间交换信息的最佳途径之一。使用对话框函数可以创建 Visual FoxPro 的内部对话框，这种方法具有操作简单及快速的特点。

MESSAGEBOX()函数在对话框中显示信息，等待用户单击按钮，并返回一个整数以标明用户单击了哪个按钮。MESSAGEBOX()函数的功能用于显示一个消息框，其格式如下所示：

```
MESS AGEBOX(CMessageText,[nDialogboxType[,CTitleBarText]])
```

其中：

- CMessageText: 表示显示在消息框中的正文内容。
- nDialogboxType: 确定消息框中要显示哪些按钮和图标，一般有 3 个参数，其取值和含义为
  - 0~5: 表示出现在消息框中的按钮，如表 2.6 所示。

表 2.6 按钮类型和数目

值	按钮
0	只有“确定”按钮
1	具有“确定”和“取消”按钮
2	具有“终止”、“重试”和“忽略”按钮
3	具有“是”、“否”和“取消”按钮
4	具有“是”和“否”按钮
5	具有“重试”和“取消”按钮

- 16, 32, 48, 64: 表示出现在消息框中的图标，如表 2.7 所示。

表 2.7 图标类型

值	图标
16	停止图标
32	问号图标
48	感叹号图标
64	信息图标

- 0, 256, 512: 表示消息框中哪些是默认按钮。

表 2.8 默认按钮

值	默认按钮
0	第一个按钮
256	第二个按钮
512	第三个按钮

例如，“1 + 32 + 256”表示消息框中有“确定”、“取消”按钮，有问号图标，“取消”按钮是默认按钮；“3 + 64 + 512”表示消息框中有“是”、“否”、“取消”按钮，有信息图标，“取消”按钮是默认按钮。如果默认为 nDialogboxType，则消息框中只有“确定”按钮。

- CTitleBarText 表示出现在消息框标题栏中的文本。如果没有设定此项，则系统给出的默认标题为 Microsoft Visual FoxPro。

MESSAGEBOX( )函数的返回值是一个数值，用于确定在消息框中选择了哪个按钮，具体的返回值信息请参见表 2.9。

表 2.9 MESSAGEBOX( )函数的返回值

返回值	图标
1	“确定”按钮
2	“取消”按钮
3	“终止”按钮
4	“重试”按钮
5	“忽略”按钮
6	“是”按钮
7	“否”按钮

提示：

- 如果省略了某些可选项，必须加入相应的逗号分隔符。
- 在程序运行的过程中，有时需要显示一些简单的信息如警告或错误等，此时可以利用对话框来显示这些内容。当用户接收到信息后，可以单击相应的按钮来关闭对话框，并返回单击的按钮值。

例如，使用 MESSAGEBOX()函数显示一个消息框，正文为“这是一个教师工资管理的数据库！”，消息框中有“是”、“否”、“取消”按钮，有信息提示图标，“取消”按钮是默认按钮，消息框标题栏中的文本为“消息框”。具体函数命令如下所示，图2.2通过窗口显示了该信息。

```
? MESSAGEBOX("这是一个教师工资管理的数据库!",3 + 64 + 512,"消息框")
```

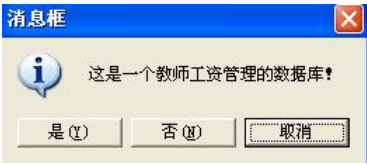


图 2.2 消息框

练习与操作

练习 1 在【命令】窗口中依次执行以下命令，查看各函数的运行结果。

```
string1="abd"
string2="ab"
? LEN(string1)
? ALLTRIM(string1)
? TYPE("string1")
? AT(string2,string1)
? AT(string1,string2)
```

练习2 在【命令】窗口中依次执行以下命令，查看各函数的运行结果。

```
nn=3245.567
kk="abcde"
jj="xyz"
xx="Visual FoxPro"
yy="数据库"
? ROUND(nn,-1)
? MOD(10,-3)
? INT(10/3)
? STR(nn,7,3)
? SUBSTR(xx-yy+kk,5,6)
? LEFT(kk+xx+jj,6)
? VAL(str(nn,8,3))
? YEAR(date( ))
? DTOC(date( ),1)
? VARTYPE("nn")
? IIF(nn<100,"大","小")
? ASC("xyz")
```

练习3 在【命令】窗口中依次执行以下命令，查看 MESSAGEBOX()函数的运行结果。

```
? MESSAGEBOX("这是一个学生成绩管理的数据库!",2+64+256,"提示框")
```

## 思考与练习 2

### 一、思考题

1. 举例说明 Visual FoxPro 有哪一种字段类型和变量类型。
2. 举例说明函数返回值的类型和函数对参数的类型要求。
3. 什么是数组？
4. 举例说明 Visual FoxPro 有哪一种逻辑运算符。
5. 举例说明 MESSAGEBOX()函数的使用。

### 二、选择题

1. 表达式  $2*3^2 + 2*8/4 + 3^2$  的值为\_\_\_\_\_。  
A. 64                      B. 31                      C. 49                      D. 22
2. 在下面的 Visual FoxPro 表达式中，不正确的是\_\_\_\_\_。  
A. {^2001-05-01 10:10:10 AM}-10                      B. {^2001-05-01}-DATE( )  
C. {^2001-05-01 10:10:10 AM}+DATE( )                      D. {^2001-05-01 10:10:10 AM}+1000
3. 已知字符串 M="12    34    ", N="    56    78    ",则执行函数 ALLTRIM (M-N)的运算结果为\_\_\_\_\_。  
A. "12    34    56    78"                      B. "12    34    56    78"  
C. "1234    56    78"                      D. "123456    78"
4. 假定系统日期是 2009 年 2 月 20 日，有如下命令：N=(YEAR(DATE( ))-1920)%100，执行后 N 的值为\_\_\_\_\_。

- A. 2009                      B. 89                      C. 1920                      D. 20
5. 设  $M = "20"$ ,  $N = "M"$ , 执行命令 `? &N+"09"` 的值是\_\_\_\_\_。
- A. 2009                      B. 20                      C. M09                      D. 出错信息
6. 职工数据库中有日期型字段“出生日期”, 要计算职工的整数年龄, 应当使用命令\_\_\_\_\_。
- A. `?DATE()-出生日期/365`                      B. `? (DATE()-出生日期)/365`  
C. `?INT((DATE()-出生日期)/365)`                      D. `?ROUND((DATE()-出生日期)/365)`
7. 关于“?”和“??”, 下列说法中错误的是\_\_\_\_\_。
- A. “?”和“??”只能输出多个同类型的表达式的值  
B. “?”从当前光标所在行的下一行第0列开始显示  
C. “??”从当前光标处开始显示  
D. “?”和“??”后可以没有表达式
8. `DTOC({^2008/09/28})` 的值应为\_\_\_\_\_。
- A. 2008 年 9 月 28 日                      B. 08/09/28  
C. 09/28/08                      D. 2008/09/28
9. 下列数据中, 不是常量的是\_\_\_\_\_。
- A. NAME                      B. “年龄”                      C. "91/01/02"                      D. .T.
10. 假定 X 为 N 型变量, Y 为 C 型变量, 则下列选项中符合 Visual FoxPro 语法要求的表达式是\_\_\_\_\_。
- A. `NOT.X >= Y`                      B. `Y*2 > 10`                      C. `X.001`                      D. `STR(X)-Y`

### 三、填空题

1. 若  $a = 5$ ,  $b = "a < 10"$ , 则 `? VARTYPE(b)` 输出的结果为\_\_\_\_\_, 而 `? VARTYPE("b")` 输出的结果为\_\_\_\_\_, `VARTYPE(&b)` 输出的结果为\_\_\_\_\_。
2. 设系统日期为 2009 年 01 月 12 日, 下列表达式的值是\_\_\_\_\_。
- `? VAL(SUBSTR("2009", 3)) + LEN(STR(YEAR(DATE()), 4)) + 17`
3. 在访问内存变量时, 必须在变量名前加上前缀\_\_\_\_\_。
4. 数组在使用时要由\_\_\_\_\_或\_\_\_\_\_命令来定义。
5. `YEAR()` 函数返回值的数据类型是\_\_\_\_\_。
6. “年龄大于 50 岁或小于 20 岁的技术员”的逻辑表达式\_\_\_\_\_。
7. `? REPLICATE("$", 6)` 的结果是\_\_\_\_\_。
8. 工资关系中有工资号、姓名、职务工资、津贴、公积金、所得税等字段, 其中可以作为关键字的字段是\_\_\_\_\_。
9. 命令 `? TYPE("10/25/08")` 的输出值是\_\_\_\_\_。
10. 设字段变量 sex 是字符型, score 是数值型, 分别存放“性别”和“成绩”信息, 若要表达“性别是男的, 并且成绩大于等于 90 分”这一命题, 表达式应是\_\_\_\_\_。

# 第 3 章 数据库的基本操作

数据库系统是组织、管理现实世界中的庞大数据的强有力手段。在 Visual FoxPro 中，一个数据库可以是多个数据表的组合，它是开发应用程序的基础，数据库文件的扩展名为.dbc。在 Visual FoxPro 中数据以表的形式存放。数据库中的一个表就是一个关系，它是用来存储数据的文件，数据表文件的扩展名是.dbf。

## 3.1 数据库与数据表

数据库是一种工作环境，一个数据库文件可以是多个数据表的组合，例如现在建立一个名为“学生”的数据库，该数据库共含两张表：“学生表”和“学生选课表”。

### 3.1.1 数据库的基本操作

在 Visual FoxPro 中，数据库包含数据库表、本地视图、远程视图、连接和存储过程。在创建数据库表、本地视图、远程视图等对象之前必须先创建一个数据库。

#### 1. 创建数据库的方法

##### 1) 在【项目管理器】窗口中创建数据库

在【项目管理器】窗口中，选择【数据】选项卡中的【数据库】，然后单击【新建】按钮，如图 3.1 所示。

在弹出的【创建】对话框中输入数据库名“学生”，单击【保存】按钮，如图 3.2 所示。这样，一个“学生”数据库已经建立。由于当前该数据库中未包含任何数据对象，所以该数据库是一个空的数据库，如图 3.3 所示。但此时【项目管理器】窗口中已经出现了“学生”数据库文件，并会看到其下方自动包含了表、本地视图、远程视图、连接和存储过程等。



图 3.1 在【项目管理器】窗口中创建数据库

##### 2) 由【文件】菜单创建数据库

在系统菜单中，选择【文件】下拉菜单中的【新建】命令，在【新建】对话框中选择数据库文件类型，然后单击【新建文件】按钮，此时系统会弹出【新建】对话框。在弹出的【新建】对话框中输入数据库名“学生”，并单击【保存】按钮，此时系统会显示【数据库设计器】窗口（如图 3.3 所示），在没有加入任何内容时同样也是一个空的数据库。

#### 2. 数据库的打开与关闭

##### 1) 数据库的打开

在对数据库进行维护之前，必须先将数据库打开。在【文件】菜单中选择【打开】命令，

或单击常用工具栏上的“打开”按钮。在弹出的对话框中，指定预打开文件所在的文件夹、类型及文件名。

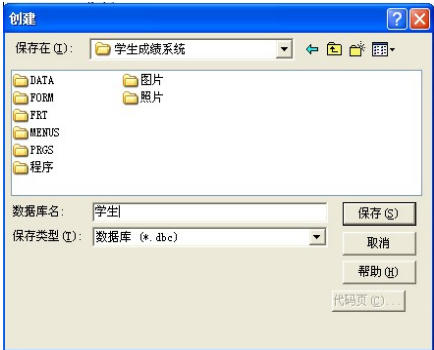


图 3.2 在【创建】对话框中输入数据库名“学生”



图 3.3 【数据库设计器】窗口

2) 数据库的关闭

使用完数据库之后要及时将其关闭。单击【数据库设计器】窗口右上角的关闭按钮即可。

除了可以利用菜单操作之外，也可以使用命令方式操作。命令方式操作将在后面集中讲述。

3. 在数据库中添加表

在【项目管理器】窗口中，选中【数据】选项卡中的【数据库】左边的“+”号，并选择其下方的【表】（如图 3.4 所示），然后单击【添加】按钮。在弹出的【打开】对话框中，选择预添加表文件所在的文件夹、类型及表文件名，如上述的“学生表”，然后单击【确定】按钮。这个“学生表”便被添加到数据库中。



图 3.4 在【项目管理器】窗口中添加数据库

在【数据库设计器】窗口中单击鼠标右键，在弹出的快捷菜单中选择【添加表】命令（如图 3.5 所示），或单击【数据库设计器】窗口工具栏上的“添加表”按钮。也可将预添加的表文件添加到数据库中。结果如图 3.6 所示。

在【数据库设计器】窗口中单击鼠标右键，在弹出的快捷菜单中选择【添加表】命令（如图 3.5 所示），或单击【数据库设计器】窗口工具栏上的“添加表”按钮。也可将预添加的表文件添加到数据库中。结果如图 3.6 所示。



图 3.5 在快捷菜单中选择【添加表】命令

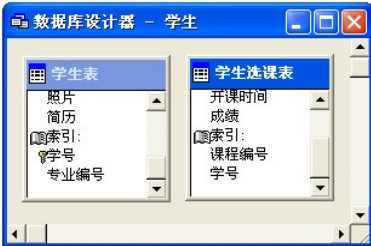


图 3.6 将表文件添加到数据库中



#### 4. 在数据库中移去/删除表

在【项目管理器】窗口中，选中【数据】选项卡中【数据库】左边的“+”号并选择其下方的“学生表”，然后单击【移去】按钮。在弹出的对话框中，选择【移去】或【删除】按钮。

在【数据库设计器】窗口中预删除的表上单击鼠标右键，在弹出的快捷菜单中选择【删除】命令，或单击【数据库设计器】窗口工具栏上的“移去表”按钮，也可将预添加的表文件从数据库中移去或删除。

#### 5. 在数据库中查找表

在【数据库设计器】窗口中单击鼠标右键，在弹出的快捷菜单中选择【查找对象】命令，在弹出的对话框中完成查找。在主窗口的【数据库】菜单中选择【查找对象】命令，也可在弹出的对话框中进行查找。

### 3.1.2 数据表的基本概念

在 Visual FoxPro 中，数据库是由表结构和表记录两部分组成，要建立一个数据表，首先需要设计和建立一个数据表结构，然后输入相应的表记录。

#### 1. 数据表

在 Visual FoxPro 中，数据表分为数据库表和自由表两种。未加入某个数据库的表称为自由表，自由表之间没有必要的关联。将一个自由表添加到某个数据库中或在数据库设计器中创建的表就成为数据库表。虽然数据库表和自由表都能存储数据，但数据库表的功能更强大。

#### 2. 数据表的命名

前面已经知道，数据表的扩展名为.dbf，创建表时还应指定一个对应于.dbf 文件的文件名。Visual FoxPro 中的表名可由字母、汉字、数字和下划线组成，但表名的第一个字符必须是字母或下划线。数据库表可以有长文件名（字符数 $\leq 128$ 个）。

为表命名时应注意，表中不应包含空格，并且表名应简明易记。

#### 3. 字段名

字段名是给每个字段定义的唯一名称，字段名可由字母、汉字、数字和下划线组成，中间不允许含有空格，数据库表字段名的长度 $\leq 128$ 个字符，自由表字段名的长度 $\leq 10$ 个字符。在同一个表文件中，不得有相同的字段名。在 Visual FoxPro 操作期间，所有操作都是通过字段名来访问字段的。

#### 4. 记录

表以类似于二维表格的形式存储数据，由若干条（行）记录组成，每条记录包含若干个

字段（列），每个记录的每个字段中存储一个 Visual FoxPro 允许使用类型的数据。通常把表的框架，也就是表由哪些字段构成称为表的结构，将表中的记录称为表的内容。

5. 字段的数据类型

与其他程序设计语言相比，Visual FoxPro 提供了更多的数据类型。字段的数据类型主要有以下 13 种，如图 3.7 所示。字段类型及其功能如表 3.1 所示。



图 3.7 字段的数据类型

表 3.1 字段类型及其功能

数据类型	中文名字	占用内存	用途	示例
Character	字符型 (C)	每个字符占 1 个字节	任何字符	书名、人名、地名等
Currency	货币型 (Y)	8 个字节	存储货币量	产品价格、货物价格等
Numeric	数值型 (N)	8 个字节 (内存中)， 1~20 个字节 (表中)	记录中的整数或者小数	数值量如成绩、课时等
Float	浮点型 (F)	与数值型相同	与数值型相同	与数值型相同
Date	日期型 (D)	8 个字节	设置月/日/年	出生日期
DateTime	日期时间型 (T)	8 个字节	设置月/日/年 时间	工作时间
Double	双精度型 (D)	8 个字节	记录双精度浮点型	实验数据
Integer	整型 (I)	4 个字节	整数	如年龄、学生人数等
Logical	逻辑型 (L)	1 个字节	真与假的布尔值	如团员否、婚否等
Memo	备注型 (M)	在表中为 4 个字节	字母、文字、数字	如个人爱好、简历等，在表中显示 memo
General	通用型 (G)	在表中为 4 个字节	存储 OLE 对象	如扩展名为.doc 的文档或位图文件等，在表中只显示 gen
Character (Binary)	字符型 (二进制)	在表中为 4 个字节	与上述的“字符型”相同，当更改代码时字符值不会改变	用于在表中保存用户密码
Memo (Binary)	备注型 (二进制)	在表中为 4 个字节	与上述的“备注型”相同，当更改代码时备注内容不变	用于存储不同国家（地区）的登录脚本

在上述数据类型中，双精度型、浮点型、整型、通用型、备注型这几个类型仅适用于字段，而其他类型也适用于内存变量。

3.1.3 数据表结构的建立

创建数据表就是建立一个新的表文件。创建数据表之前，需要对表进行分析，确定准备在表中存储哪些数据。

数据表的创建可分两步进行。首先，创建数据表的结构，即确定数据表的字段个数、字段名、字段类型、字段宽度及小数位数等特征；其次，根据字段特征输入相应的记录。

例如，现在要创建一个如表 3.2 所示的“学生表”，首先需要分析表中的相关数据。

表 3.2 学生表

学号	姓名	专业编号	性别	出生日期	入学成绩	团员否	照片	简历
080301001	张跃林	03	男	1989-10-23	589	T	—	—
080301020	张文斌	03	男	1989-5-20	593	T	—	—
080302045	陈江城	03	男	1989-8-4	598	T	—	—
084201002	夏利华	42	女	1989-9-4	497	F	—	—
074202123	李林萍	42	女	1989-3-5	516	T	—	—
...	...	...	...	...	...	...	...	...

从表 3.2 可以发现，该表共有 9 个字段和相对应的记录。要建立该数据表，首先要确定每个字段的属性，即字段名、数据类型、宽度、小数位数，然后就可以建立表结构了。表结构如表 3.3 所示，表的记录如表 3.2 所示。

表 3.3 “学生表”的结构

字段名	数据类型	宽度	小数位数
学号	字符型	9	
姓名	字符型	6	
专业编号	字符型	2	
性别	字符型	2	
出生日期	日期型	8	
入学时间	日期型	8	
入学成绩	数值型	5	1
团员否	逻辑性	1	
照片	字符型	4	
简历	备注型	4	

在 Visual FoxPro 中，系统提供了多种创建表结构的方法。可以通过表向导创建表结构，或者通过表设计器创建表结构，或者通过在【命令】窗口输入 CREAT 命令来创建表结构。

1. 利用表向导建立表结构

可以利用表向导创建表结构。由于向导中有一步步的操作提示，对初学者来说简单、方便。现在就用表向导来创建新表，具体步骤如下。

(1) 打开表向导。在【项目管理器】窗口的【数据】选项卡下选择【自由表】后，单击【新建】按钮。在打开的【新建表】对话框中单击【表向导】按钮，如图 3.8 所示。或者选

择系统【文件】菜单下的【新建表】命令，在【新建】对话框中选择【表】，然后单击【向导】按钮。

(2) 选择字段。表向导的步骤 1 为字段选取，如图 3.9 所示。



图 3.8 【新建表】对话框

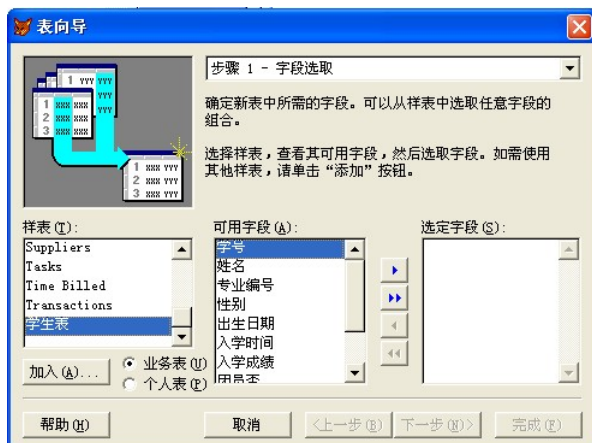


图 3.9 步骤 1-字段选取

在【表向导】窗口里的【样表】列表框中显示的是一些典型的表模板。一旦选择了一个样表，则这个表的所有字段都显示在【可用字段】列表框内。可选【业务表】或【个人表】中列出的样表，【加入】按钮用于加入样表中没有列出的表。这里单击【加入】按钮，在【打开】对话框中选取所需要“学生表”。在【可用字段】列表框内选择所有字段，然后可以进入下一步操作。

图 3.10 为表向导的步骤 1a，该步骤用于选择是创建自由表还是创建数据库表。这里选择自由表，单击【下一步】按钮。

(3) 修改字段设置。表向导的步骤 2 为修改字段设置，如图 3.11 所示。在这个对话框中可以浏览和修改字段的属性，改变字段的宽度。单击【类型】组合框会出现一个下拉列表，其中列出了所有字段的数据类型。可在这个对话框中改变字段名。设置完成后，单击【下一步】按钮。

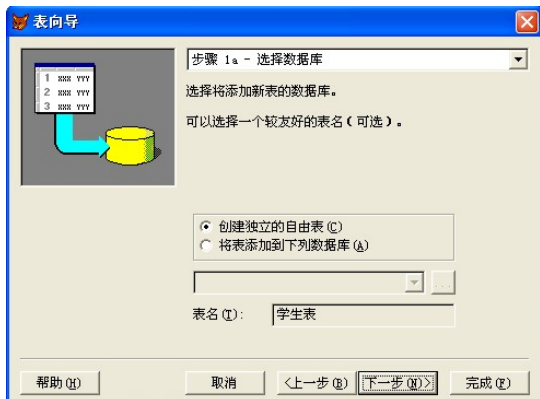


图 3.10 步骤 1a-选择数据库



图 3.11 步骤 2-修改字段设置

(4) 建立索引。表向导的步骤3即为表建索引,如图3.12所示。在这个对话框中可以为表设置索引。设置完后,单击【下一步】按钮。

(5) 完成。表向导的步骤4即完成表设置,如图3.13所示。这是表向导的最后一步。选择了创建表后的出口(三个选项任选一项)后,单击【完成】按钮,在弹出的【另存为】对话框中,输入表名,选择正确路径后,单击【保存】按钮。至此,一个完整的数据表就建成了。



图 3.12 步骤3-为表建索引

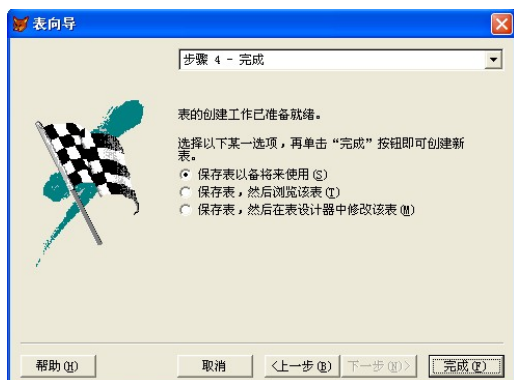


图 3.13 步骤4-完成

## 2. 利用表设计器建立表结构

在【项目管理器】窗口中选择【自由表】,然后单击【新建】按钮,可以用表设计器直接创建一张新表。具体步骤如下。

(1) 打开表设计器。打开表设计器的方法与打开表向导的方法很相似,主要有以下两种:

- 在【项目管理器】窗口的【数据】选项卡下选择【自由表】后,单击【新建】按钮。在打开的【新建表】对话框中单击【新建表】按钮。
- 选择【文件】菜单下的【新建】命令,在【新建】对话框中选择【表】,然后单击【新建文件】按钮。通过以上两种方法均可打开表设计器,如图3.14所示。

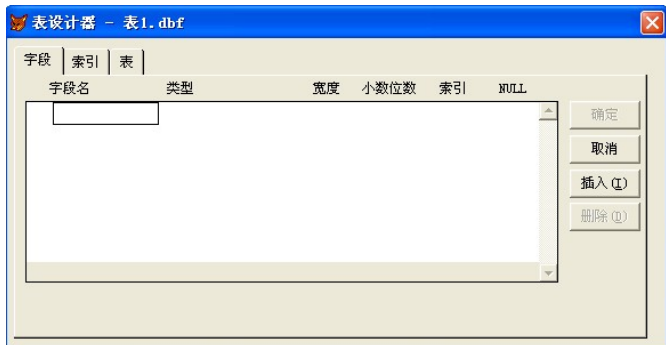


图 3.14 表设计器

(2) 输入内容。打开表设计器后,便可按要求输入字段名、字段类型、宽度、小数位和索引等。

### 3.1.4 数据表结构的操作

#### 1. 将数据表添加到项目中

如果已经存在多个扩展名为.dbf 的表，可以将这些表添加到项目中。

##### 1) 将数据库表添加到项目中

在【项目管理器】窗口的【数据】选项卡中，选择指定数据库下的表，单击【添加】按钮，在弹出的对话框中选择待添加到项目中的表，单击【确定】按钮即可，如图 3.15 所示。

##### 2) 将自由表添加到项目中

在【项目管理器】窗口的【数据】选项卡中，选择【自由表】，单击【添加】按钮，在弹出的对话框中选择待添加到项目中的自由表，单击【确定】按钮即可。

#### 2. 添加/删除字段

建立表结构后，需要时还可以向表中添加新的字段。例如，可以按下面的步骤向【学生表】中添加两个新的字段。

(1) 在【项目管理器】窗口的【数据】选项卡中，选择要修改的“学生表”，如图3.16 所示。



图 3.15 将数据库表添加到项目中

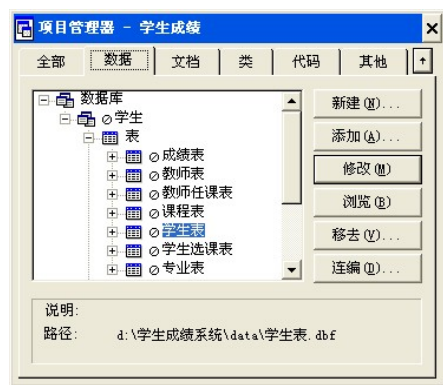


图 3.16 选择要修改的表

(2) 单击【修改】按钮，进入表设计器；单击“专业编号”字段，然后单击【插入】命令按钮，在“专业编号”字段前插入了一个新的字段。在【字段名】栏中输入“籍贯”，在【类型】栏中选择“字符型”，在【宽度】栏中选择“6”。

(3) 单击“照片”字段，然后单击【插入】命令按钮，在“照片”字段前插入了一个新的字段。在【字段名】栏中输入“班级”，在【类型】栏中选择“字符型”，在【宽度】栏中选择“6”。

(4) 然后单击【确定】按钮。在弹出的对话框提示是否永久更改表结构时，单击【是】按钮。

(5) 对于表中多余的字段，可在表设计器中选择待删除的字段，然后单击【删除】按钮，如图 3.17 所示。

### 3. 修改字段

建立表结构后,当字段的某些属性无法满足实际需要时,就必须进行修改。例如,可以按照下面的步骤在“学生表”中修改相应的字段属性。

在【项目管理器】窗口的【数据】选项卡中,选择“学生表”;单击【修改】按钮,进入表设计器;单击“姓名”字段,在【宽度】栏中将字段宽度设为 10,然后单击【确定】按钮。在弹出的对话框提示是否永久更改表结构时,请单击【是】按钮。这样就可以修改表的字段。

### 4. 调整字段顺序

利用鼠标选定某字段左边的灰色方块,拖动鼠标向上或向下移动到新位置后释放鼠标。例如,将“出生日期”字段移动到“照片”字段上,先选择“出生日期”字段左边的灰色方块,然后将其拖动至“照片”字段处释放即可,如图 3.18 所示。修改完后,单击【确定】按钮返回【项目管理器】窗口。

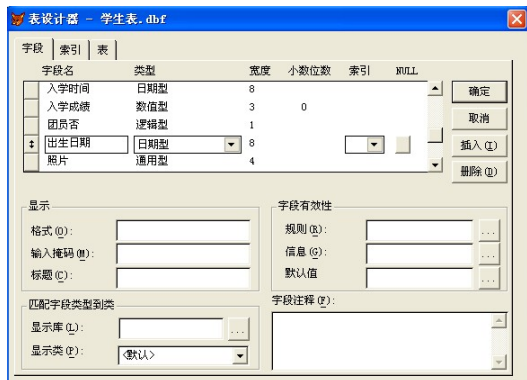


图 3.17 添加/删除字段



图 3.18 调整字段顺序

### 5. 设置字段属性

数据库表和自由表都能存储数据,但数据库表的功能更强大。当一个数据表隶属于数据库成为数据库表后,该数据库表就可以使用长文件名和长字段名( $\leq 128$ 个字符)。表中的字段可以有标题和注释,可设置字段级规则和记录级规则,也可设置触发器和永久关系等。

### 6. 为字段添加注释

在 Visual FoxPro 中,可以为数据库表中的字段增加注释,以便更详细地描述一个字段所代表的含义。下面以“学生表”为例,简述为字段添加注释的步骤。

- (1) 在【项目管理器】窗口中,选择“学生表”,单击【修改】按钮,打开表设计器。
- (2) 在【表设计器】窗口中,单击要添加注释的字段,如“专业编号”字段。
- (3) 在【字段注释】区域中,输入“学生现设专业二年后会有所调整”,如图 3.19 所示。然后单击【确定】按钮,这时系统会显示提示对话框。

(4) 在提示对话框中再次单击【是】按钮,关闭【表设计器】窗口。此时,系统已将添加的注释永久性地保存到表结构中。



7. 设置字段标题

Visual FoxPro 允许为数据库表中的字段设置字段标题，以便更好地理解字段的含义，增强字段的可读性，在浏览的窗口中，字段标题将代替原字段名。下面以“学生表”为例，简述设置字段标题的步骤。

- (1) 在【项目管理器】窗口中，选择“学生表”，单击【修改】按钮，打开表设计器。
- (2) 在【表设计器】窗口中，单击要添加标题的字段，如“学号”字段。
- (3) 在下方【显示】区域的【标题】文本框中，输入“学生学号”，如图3.20所示。然后单击【确定】按钮，这时系统会显示提示对话框。
- (4) 在提示对话框中再次单击【是】按钮，关闭【表设计器】窗口。此时，系统已将添加的标题永久性地保存到表结构中了。



图 3.19 为字段添加注释



图 3.20 设置字段标题

8. 设置字段有效性规则

在为字段输入数据时，经常会遇到误输入数据的情况。在 Visual FoxPro 中，提供了为数据库表中的字段设置有效性规则的功能，利用此规则判断输入的数据是否符合此字段的实际意义。例如，在“学生表”中，输入“入学成绩”字段值时，其值应在 550~750 之间，如果所输入的数据超出了这个范围，说明此数据是无效的，操作步骤如下。

- (1) 在【项目管理器】窗口中，选择“学生表”，单击【修改】按钮，打开表设计器。
- (2) 在【表设计器】窗口中，单击要建立规则的字段，如“入学成绩”字段。
- (3) 单击下方【字段有效性】区域的【规则】文本框右侧的浏览按钮 (...)，打开【表达式生成器】对话框。在【规则】文本框中输入“BETWEEN (入学成绩, 550, 750)”，如图 3.21 所示。然后单击【确定】按钮，关闭【表达式生成器】对话框，返回到【表设计器】窗口。
- (4) 在【信息】文本框中输入“取值范围在 550~750 之间，请重新输入正确的数值”。然后单击【确定】按钮，这时系统会显示提示对话框。

在提示对话框中再次单击【是】按钮，关闭【表设计器】窗口。此时，系统已将设置的字段规则永久性地保存到表结构中。当“入学成绩”字段输入的数据不符合要求时，屏幕会显示出错信息，如图 3.22 所示。





图 3.21 设置字段有效性规则



图 3.22 Visual FoxPro 显示的错误信息

## 3.1.5 数据表记录的操作

### 1. 输入记录

建立表结构后，便可以向表中输入记录了。输入记录有三种最常见的情况。

① 情况一：在【表设计器】窗口建好表结构后，单击【确定】按钮，系统弹出对话框，如图3.23所示。在弹出的对话框中，单击【是】按钮，就会进入表的浏览界面，如图3.24所示，便可输入记录了。

② 情况二：保存了表的结构后，再想输入表的记录，可在【项目管理器】窗口中选择数据表（数据库表或自由表）名，单击【项目管理器】窗口右边的浏览按钮，就会进入表的浏览界面，随后可以输入记录。



图 3.23 系统输入记录提示框

③ 情况三：打开已保存的表文件，选择系统菜单中的【显示】→【浏览】→【浏览学生表（数据库！学生表）】命令，打开浏览界面，随后可以输入记录。

例如，在项目管理员的【数据】选项卡中，选择数据表（数据库表或自由表）；单击浏览按钮，即可打开浏览窗口，如图3.24所示。此时也可选择系统菜单中的【显示】→【编辑】命令，即可打开编辑窗口，如图3.25所示。选择【显示】菜单中的【追加记录】命令，即可输入相应的记录内容。一个字段输入完毕后，按下 Tab 键即可将光标移到下一个字段；输入完一条记录后，按下向下方向键即可将光标定位到下一行。输入完毕后单击窗口右上角的关闭按钮，系统会自动存盘退出。

### 2. 输入记录的方法

在输入记录时必须根据表结构相对应的字段属性输入相关的记录，各类型字段输入数据的方法有所不同，下面来看一下向记录中各类型字段输入数据的方法。

- ① 对于常用的字符型字段数据，可按其字段属性的要求，输入字符或汉字即可。
- ② 对于数值型、货币型、浮点型、双精度型和整型字段数据，应采用十进制日常表示

法输入。如果输入数据太大或太小，超出了定义的字段宽度所能表示的范围，则显示为若干个“\*”号。对于逻辑型字段，在输入数据时，若想输入 t，可以键入字母 t 或 y；若想输入 f，可以键入字母 f 或 n。输入时不区分大小写。

学号	姓名	专业编号	性别	出生日期	入学时间	入学成绩	团员否	照片	简历
080301001	张跃林	03	男	10/23/83	09/01/08	589	T	Gen	Memo
080301020	张文斌	03	男	05/20/84	09/01/08	593	T	Gen	memo
080302045	陈江城	03	男	08/04/83	09/01/08	598	T	Gen	Memo
070403128	赵晓飞	04	男	04/06/83	09/01/07	521	F	Gen	memo
080401081	孟江浩	04	男	09/20/84	09/01/08	623	T	Gen	memo
070401032	刘慧琴	04	女	07/30/83	09/01/07	599	T	Gen	memo
084104112	郝明雅	41	女	10/31/83	09/01/08	471	F	Gen	memo
074101067	李巧兰	41	女	05/03/83	09/01/07	473	F	Gen	memo
084201002	夏利华	42	女	09/04/83	09/01/08	497	F	Gen	memo
074202123	李林萍	42	女	03/05/84	09/01/07	516	T	Gen	memo
084205255	李雅迪	42	女	07/08/83	09/01/08	478	T	Gen	memo
084202152	殷 俐	42	女	01/02/83	09/01/08	506	T	Gen	memo

图 3.24 表的浏览界面

学生表

学号 080301001

姓名 张跃林

专业编号 03

性别 男

出生日期 10/23/83

入学时间 09/01/08

入学成绩 589

团员否 T

照片 Gen

简历 Memo

---

学号 080301020

姓名 张文斌

专业编号 03

性别 男

出生日期 05/20/84

入学时间 09/01/08

入学成绩 593

团员否 T

照片 Gen

图 3.25 表的编辑界面

③ 通用型字段数据的记录处显示为 gen。鼠标双击 gen 或将光标停留在通用字段处按下 Ctrl+PgDn 组合键，出现通用型字段的编辑窗口。选择【编辑】菜单中的【插入对象】命令，在弹出的对话框中选择预插入的文件，单击【确定】按钮。单击通用型字段编辑窗口的关闭按钮，结束对通用型字段的编辑。此时可以看到 gen 已经变成了 Gen。除此之外，利用 Windows 中的剪贴板，可以把从其他程序中复制的图片通过【编辑】菜单的【粘贴】命令粘贴到通用型字段的编辑窗口中。

④ 备注型字段数据的记录处显示为 memo。鼠标双击 memo 或将光标停留在备注型字段处按下 Ctrl+PgDn 组合键，出现备注型字段的编辑窗口。在该窗口输入相应的数据，数据输入完毕后单击编辑窗口右上角的关闭按钮（或按 Ctrl + F4 组合键），结束对备注型字段的编辑。此时可以看到 memo 已经变成了 Memo 了。此外，在编辑过程中如果想放弃输入的内容，可以按下 Esc 键，撤销刚才的操作。

⑤ 在日期型字段及日期时间型字段中，日期的输入格式可用 set date 命令进行设置，默认为美国日期格式（月/日/年，即 mm/dd/yy），其中年份最好输入 4 位。

⑥ 如果某字段允许接受 NULL 值，就可以使用 Ctrl+0 组合键向字段中输入 NULL 值。

3. 追加记录

1) 在【显示】菜单下追加记录

在浏览或编辑窗口，选择系统菜单的【显示】→【追加记录】→【追加方式】命令。系统会在表的末尾添加一条空记录，并显示一个输入框。当输入完一条记录后，系统会自动追加下一条记录。如想解除追加方式，可按 Ctrl + W 组合键保存修改，或按 Ctrl + Q 组合键放弃修改来关闭浏览窗口。

2) 在【表】菜单下追加记录

在浏览或编辑窗口，选择系统菜单的【表】→【追加新记录】命令。系统会在表的末尾添加一条空记录，并显示一个输入框，如图 3.26 所示。这种方式只允许添加一条记录，也就

是说,输入完一条记录后,系统不会自动追加下一条记录,若想再添加一条记录,必须再选择一次【追加新记录】命令。



教师编号	教师姓名	性别	职称	毕业学校	课程编号	工资	政府津贴
JC0101	陈得江	男	教授	北京大学	CC01	3310.10	T
JC0202	赵惠兰	女	副教授	清华大学	CC01	2587.00	F
JC0303	刘大林	男	讲师	上海交通大学	CC02	2000.00	F
JS0101	秦健中	男	副教授	浙江大学	CS01	3120.00	T
JS0202	吴莲敏	女	副教授	复旦大学	CS01	1500.00	F
CC0101	赵晓兰	女	教授	同济大学	CS02	3334.00	F
CC0202	李月敏	女	讲师	吉林大学	CS02	1878.00	F
CC0303	王克强	男	讲师	北京理工大学	CC03	1564.00	F
CC0404	刘雪燕			上海交通大学	CS01	2345.00	F
CS0101	林黎明			京林业大学	CS02	3254.00	F
CS0202	赵华林			门大学	CS02	2786.00	F
CS0303	季严冬	男	教授	哈尔滨工业大学	CC03	3786.00	T

图 3.26 在【表】菜单下追加记录

### 3) 追加符合条件的一组记录

在浏览或编辑窗口,选择【表】菜单下的【追加记录】选项,系统会显示【追加来源】对话框,如图 3.27 所示。

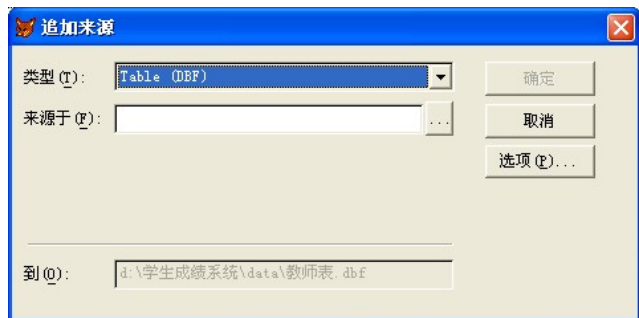


图 3.27 【追加来源】对话框

其中各选项的含义如下所示:

- 类型: 用于选择来源的数据类型。
- 来源于: 用于指定数据来源。
- 选项: 用于选择符合条件的数据。

选择 Table (表) 类型后,单击【来源于】文本框右侧的浏览按钮,出现【打开文件】对话框,在此选择需要的数据所在的表,然后单击【确定】按钮。如果不需要选定表中的所有数据,可以单击【选项】按钮,打开【追加来源选项】对话框,如图 3.28 所示。单击【字段】按钮,可打开【字段选择器】对话框,用于选择所需字段。用户可直接输入筛选表达式,也可以单击左侧的【For】按钮,在打开的【表达式生成器】窗口生成一个表达式。确定好类型、来源和选项后,单击【确定】按钮,系统会自动把选定的表中符合条件的记录添加到当前所在表的末尾。这种方式可从选定文件中向当前所在的表中一次添加多条记录。



图 3.28 【追加来源选项】对话框

## 4. 修改字段中的记录

在数据表中，可以通过浏览窗口或者编辑窗口来编辑相应的数据。在系统的【显示】菜单中选择【浏览】或【编辑】命令，进入表的浏览窗口或编辑窗口。在浏览窗口中，一行显示一条记录，在编辑窗口中则一行显示一个字段。

在修改字符型、数值型、逻辑型、日期型等字段时，可以进行如下的操作。

① 插入新数据。鼠标单击要插入数据的字段，出现闪烁的插入点，使用左右方向键移动插入点至适当的位置，输入新的数据。

② 修改部分数据。将插入点置于要修改的字段中，或用鼠标拖动的方式选定要修改的数据，删除、覆盖旧的数据，输入新的数据。

③ 修改字段中全部的数据。将光标定位在要修改的字段上，选择【编辑】菜单中的【全部选定】命令或通过鼠标拖动的方式选定全部数据，键入新的数据替换旧的数据。另外，也可以通过 Tab 键或方向键将其定位到该字段，键入新的数据替换旧的数据。

④ 对于通用型字段的修改，可在浏览窗口中双击“gen”，出现通用型字段编辑窗口，在该窗口中可以清除已经插入的对象，也可以重新嵌入或链接新的 OLE 对象。

⑤ 对于备注型字段，同样可在浏览窗口中双击“memo”，出现备注型字段的编辑窗口，在该窗口中可以进行数据的修改。修改完毕后单击关闭按钮结束操作或按 Esc 键放弃所做的修改。

所有的修改操作结束后，单击浏览数据窗口右上角的关闭按钮即可保存修改。

## 5. 记录的删除与恢复

Visual FoxPro 提供了逻辑删除和物理删除两种删除方式。逻辑删除是为了防止误删除操作，只在要删除的记录前加一个黑色的删除标记，这一部分记录虽然不参与任何操作，但仍存在于数据表内，万一发生误删除操作，还可将这部分记录恢复。物理删除是将记录真正地删除，数据表中不再保留这些记录，而且删除后不能恢复。

### 1) 记录的逻辑删除

当不再需要数据表的某些记录时，可以直接从表中删除这些记录。下面以“学生表”为例，介绍直接从表中删除记录的方法。

(1) 在【项目管理器】窗口的【数据】选项卡中选择“学生表”，单击【浏览】按钮。

(2) 在浏览窗口中定位要删除的记录，比如第 2 条记录。

(3) 单击该记录左边的空白方框，即可加上一个黑色的删除标记。用此方法可以给多条记录加上逻辑删除标记，如图 3.29 所示。

学号	姓名	专业编号	性别	出生日期	入学时间	入学成绩	团员否	照片	简历
080301001	张跃林	03	男	10/23/83	09/01/08	589	T	Gen	Memo
080301020	张文斌	03	男	05/20/84	09/01/08	593	T	Gen	memo
080302045	陈江城	03	男	08/04/83	09/01/08	598	T	Gen	Memo
084201002	夏利华	42	女	09/04/83	09/01/08	497	F	Gen	memo
074202123	李林萍	42	女	03/05/84	09/01/07	516	T	Gen	memo
070403128	赵晓飞	04	男	04/06/83	09/01/07	521	F	Gen	memo
084104112	郝明雅	41	女	10/31/83	09/01/08	471	F	Gen	memo
084205255	李雅迪	42	女	07/08/83	09/01/08	478	T	Gen	memo
080401081	孟江浩	04	男	09/20/84	09/01/08	623	T	Gen	memo
070401032	刘慧琴	04	女	07/30/83	09/01/07	599	T	Gen	memo
084202152	殷 俐	42	女	01/02/83	09/01/08	506	T	Gen	memo
074101067	李巧兰	41	女	05/03/83	09/01/07	473	F	Gen	memo

图 3.29 给记录加上逻辑删除标记

## 2) 删除符合条件的一组记录

在数据表中，如果一次要删除一组记录，或是删除符合条件的记录时，可以利用【表】菜单下的【删除记录】命令来实现。在“学生表”浏览窗口中，选择【表】菜单下的【删除记录】命令，出现【删除】对话框，如图 3.30 所示。图中的【作用范围】组合框表示要删除的记录的范围，单击其右边的下拉箭头，在打开的下拉列表框中有 4 种选择范围。

① A11：表示删除全部记录。

② Next：表示删除从当前记录开始往下的一定数目的记录（包括当前记录），记录个数由右边文本框中的数字来决定。如图 3.31 所示，表示删除从当前记录往下的 4 条记录（包括当前记录）。

③ Record：表示具体删除哪一条记录，记录号同样由右边的文本框设定。

④ Rest：表示删除从当前记录开始到文件尾的所有记录（包括当前记录）。

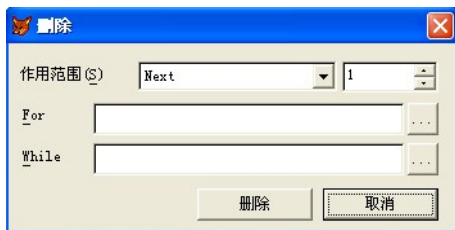


图 3.30 【删除】对话框



图 3.31 删除范围示例

在【For】文本框中设定删除条件，即可以删除在指定作用范围内满足条件的所有记录。例如，要在“学生表”中删除所有女生的记录，具体操作步骤如下。

(1) 首先将“作用范围”设定为 A11。

(2) 单击【For】文本框右边的浏览按钮，出现【表达式生成器】对话框。在【表达式】文本框中输入：性别 = "女"。

(3) 若条件设置正确，则单击【确定】按钮，返回【删除】对话框，出现的条件设定如图 3.32 所示。

(4) 单击【删除】对话框中的【删除】按钮，则系统自动删除符合指定条件的记录。在“学生表”中，共有 7 条符合条件的记录被加上了逻辑删除标记，如图 3.33 所示。





图 3.32 设置 For 条件

学生表										
	学号	姓名	专业编号	性别	出生日期	入学时间	入学成绩	团员否	照片	简历
	080301001	张跃林	03	男	10/23/83	09/01/08	589	T	Gen	Memo
	080301020	张文斌	03	男	05/20/84	09/01/08	593	T	Gen	memo
	080302045	陈江城	03	男	08/04/83	09/01/08	598	T	Gen	Memo
	084201002	夏利华	42	女	09/04/83	09/01/08	497	F	Gen	memo
	074202123	李林萍	42	女	03/05/84	09/01/07	518	T	Gen	memo
	070403128	赵晓飞	04	男	04/06/83	09/01/07	521	F	Gen	memo
	084104112	郝明雅	41	女	10/31/83	09/01/08	471	F	Gen	memo
	084205255	李雅迪	42	女	07/06/83	09/01/08	478	T	Gen	memo
	080401081	孟江浩	04	男	09/20/84	09/01/08	623	T	Gen	memo
	070401032	刘慧琴	04	女	07/30/83	09/01/07	599	T	Gen	memo
	084202152	殷 俐	42	女	01/02/83	09/01/08	506	T	Gen	memo
	074101067	李巧兰	41	女	05/03/83	09/01/07	473	F	Gen	memo

图 3.33 删除符合条件的记录

当把记录加上删除标记之后，这些带有删除标记的记录将不再参与数据表的操作，但仍然保留在数据表中，在必要的时候，可以恢复这部分记录。

【While】文本框也是用来设定删除条件的，但它只删除指定范围内连续满足此条件的记录。也就是说，当条件为真（即满足条件）时，才进行删除操作；如果条件为假（即不满足条件），便中止删除操作，不再考虑其他满足条件的记录。

3) 记录的物理删除

当确定不再需要带有逻辑删除标记的记录时，为了提高数据库的使用效率，可以将这些记录进行物理删除，彻底移出数据库。

在上述例子中，已经对一些不需要的记录打上了删除标记，若要将这部分记录物理删除，可选择【表】菜单中【彻底删除】命令。在如图3.34 所示的对话框中，单击【是】按钮，即可对一些这部分记录进行物理删除操作。通过浏览操作可以观察删除操作后的结果，如图 3.35 所示。



图 3.34 确认删除对话框



学号	姓名	专业编号	性别	出生日期	入学时间	入学成绩	团员否	照片	简历
080301001	张跃林	03	男	10/23/83	09/01/08	589	T	Gen	Memo
080301020	张文斌	03	男	05/20/84	09/01/08	593	T	Gen	memo
080302045	陈江城	03	男	08/04/83	09/01/08	598	T	Gen	Memo
070403128	赵晓飞	04	男	04/06/83	09/01/07	521	F	Gen	memo
080401081	孟江浩	04	男	09/20/84	09/01/08	623	T	Gen	memo

图 3.35 物理删除后的结果

#### 4) 恢复被逻辑删除的记录

带有逻辑删除标记的记录实际上并未被真正删除，因此只需将删除标记取消，就可以恢复被删除的记录。下面有几种恢复被逻辑删除的记录的方法。

① 直接从表中恢复记录。在数据表的浏览窗口中，使用鼠标单击记录的删除标记，即可取消删除标记，黑色的方框此时已消失。

② 恢复删除的一组记录。要恢复删除的一组记录，或者恢复符合条件的多条记录时，可以打开需要恢复表中记录的数据表，选择【表】菜单下的【恢复记录】命令，出现【恢复记录】对话框，如图 3.36 所示。该对话框的设置方法与上述的【删除】对话框完全相同，输入设置条件后单击【恢复记录】按钮即可。



图 3.36 【恢复记录】对话框

### 6. 记录的定位

在 Visual FoxPro 中，如果数据表中的记录较多，在浏览这个数据表时，假如只利用键盘、滚动条将记录指针定位到要查看的记录上会相当麻烦。Visual FoxPro 中提供了【转到记录】命令，可以实现记录指针的快速定位。

在表的浏览窗口，选择【表】菜单中的【转到记录】命令，可以拉出下一级子菜单，其中各选项的含义如下所示：

- 第一个指针移到数据表中的第一条记录上。
- 最后一个指针移到数据表中的最后一条记录上。
- 下一个指针移到当前记录的下一条记录上。
- 前一个指针移到当前记录的前一条记录上。
- 记录号指针将移到记录号所指定的记录上，选择该项时将出现【转到记录】对话框，如图 3.37 所示。
- 定位指针将移到符合条件的第一条记录上。选择该项时将出现【定位记录】对话框，该对话框的设置与前面介绍的【删除】对话框相同，如图 3.38 所示。设置完毕，单击【定位】按钮，系统立即查找符合条件的第一条记录并把记录指针移到该记录上。

### 7. 记录的查找

在 Visual FoxPro 中，如果数据表中的记录较多，在浏览数据表中的某个记录时，假如只利用键盘、滚动条将记录指针定位到要查看的记录上会相当麻烦，可以利用 Visual FoxPro 提供的工具，实现记录的快速查找。



图 3.37 【转到记录】对话框

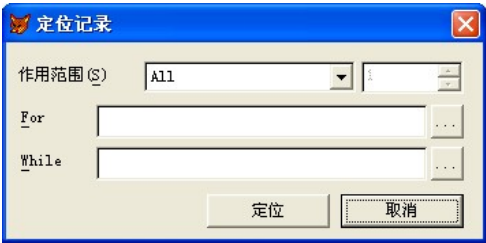


图 3.38 【定位记录】对话框

1) 利用【定位】命令

在【定位记录】对话框中设置查询条件，系统查找符合条件的第一条记录并把记录指针移到该记录上。例如，在“学生表”中查找学生专业编号是“04”的学生记录。可在【定位记录】对话框中设置定位条件：专业编号="04"，然后单击【定位】按钮，如图3.39所示。这样记录指针将定位到专业编号是 04 的学生记录上，如图 3.40 所示。



图 3.39 【定位记录】对话框

学生表										
序号	姓名	专业编号	性别	出生日期	入学时间	入学成绩	团员否	照片	简历	
080301001	张跃林	03	男	10/23/83	09/01/08	589	T	Gen	Memo	
080301020	张文斌	03	男	05/20/84	09/01/08	593	T	Gen	Memo	
080302045	陈江城	03	男	08/04/83	09/01/08	598	T	Gen	Memo	
084201002	夏利华	42	女	09/04/83	09/01/08	497	F	Gen	Memo	
074202123	李林萍	42	女	03/05/84	09/01/07	516	T	Gen	Memo	
070403128	赵晓飞	04	男	04/06/83	09/01/07	521	F	Gen	Memo	
084104112	郝明雅	41	女	10/31/83	09/01/08	471	F	Gen	Memo	
084205255	李雅迪	42	女	07/08/83	09/01/08	478	T	Gen	Memo	
080401081	孟江浩	04	男	09/20/84	09/01/08	623	T	Gen	Memo	
070401032	刘慧琴	04	女	07/30/83	09/01/07	599	T	Gen	Memo	
084202152	殷 俐	42	女	01/02/83	09/01/08	506	T	Gen	Memo	
074101067	李巧兰	41	女	05/03/83	09/01/07	473	F	Gen	Memo	

图 3.40 显示结果

2) 利用【查找】对话框

利用 Visual FoxPro 提供的查找功能，可以在数据表中查找特定的数据。例如，在“学生表”中查找学生中学号是“070401032”的学生记录，具体的操作步骤如下。

- (1) 在浏览窗口中打开“学生表”。
- (2) 选择【编辑】菜单中的【查找】命令，出现【查找】对话框。
- (3) 在【查找】对话框中输入“070401032”，如图 3.41 所示。



图 3.41 【查找】对话框



(4) 设置查找的“选项”，共有4种选择：

- 区分大小写：查找时要区分字母的大小写。
- 全字匹配：要求与查找字符串的内容完全相同才符合查找条件。
- 回绕：指当查找到表头或表尾时，重新查找。
- 向后搜索：默认的方式为从前往后的顺序查找，选中该项表示反向查找。

(5) 单击【查找下一个】按钮，开始查找指定的数据。若查找到第一个符合条件的数据，会用一个蓝色方框指定。

(6) 再次单击【查找下一个】按钮，会继续查找下一个符合设定条件的数据。重复此过程可以将所有指定的数据查找出来。直至系统发出警告声，状态栏显示“没有找到”信息。

## 练习与操作

**练习1** 使用菜单建立一个名为“数据库”的数据库文件，具体操作步骤如下。

- (1) 选择菜单栏中的【文件】菜单中的【新建】命令。
- (2) 在打开的【新建】对话框中选择“数据库”文件类型，然后单击【新建文件】按钮，此时系统会打开【创建】对话框。
- (3) 在【创建】对话框中选取保存位置（D:\学生成绩系统）和保存类型（数据库），在【数据库名】文本框中，输入建立的数据库名称——“数据库”，并单击【保存】按钮，此时系统会显示【数据库设计器】窗口。

这时建立的数据库里没有任何内容，是一个空的数据库。在D:盘的“学生成绩系统”文件夹下可以浏览到三个文件——数据库.dbc、数据库.dct、数据库.dcx（图像文档）。

**练习2** 创建学生成绩系统中的各种数据表，创建教师表的步骤如下。

- (1) 在创建表之前，先打开需要建立表的数据库，单击菜单栏的【文件】→【打开】命令，选择D:盘“学生成绩系统”文件夹下的“数据库”，文件类型为“\*.dbc”，单击【确定】按钮。
- (2) 创建表结构。先启动表设计器，选择【文件】菜单中的【新建】命令，弹出【新建】对话框。选择【表】文件类型，单击【新建文件】按钮，弹出【创建】对话框。在【输入表名】文本框中，输入建立的数据库表名称“教师表”，单击【保存】按钮，弹出表设计器。按表3.4的要求输入相关字段。

表 3.4 “教师表”的结构

字段名	类型	宽度	小数位数	索引
教师编号	字符型	4		主索引
教师姓名	字符型	6		
性别	字符型	2		
职称	字符型	8		
毕业学校	字符型	16		
课程编号	字符型	4		
工资	数值型	7	2	
政府津贴	逻辑型	2		

(3) 输入记录。按表 3.5 的要求输入相应的记录，其他表按照相应的要求参考以上步骤建立。

表 3.5 “教师表”记录

教师编号	教师姓名	性别	职称	毕业学校	课程编号	工资	政府津贴
JC0101	陈得江	男	教授	CC01	CC01	3310.00	T
JC0202	赵惠兰	女	副教授	CC01	CC01	2587.00	F
JC0303	刘大林	男	讲师	CC02	CC02	2000.00	F
JS0101	秦健中	男	副教授	CS01	CS01	3120.00	T
JS0202	吴莲敏	女	讲师	CS01	CS01	1500.00	F
...	...	...	...	...	...	...	...

## 3.2 数据表的索引与筛选

### 3.2.1 建立索引

在数据表中，为了使用户能够快速准确地查找到需要的记录，往往需要根据数据表中某些字段的值，给数据表建立一个逻辑顺序的索引文件，根据索引文件重新排列表中显示的记录，以此提高数据记录的查找效率。

建立一个索引文件时，表中记录的物理存储顺序保持不变，实际上它存储的是根据某一字段的值进行逻辑排序的一组记录号。

索引是对数据表中某一特定的字段或表达式按照一定的规则进行排序，并根据排序的结果建立索引文件。在 Visual FoxPro 中，建立索引文件实际上是建立一个包含指向.dbf 文件记录的指针文件，通过指针建立起索引文件和原数据表的对应关系。索引文件和表文件是分开的，但是索引文件不能独立使用，必须与原数据表一起工作。

#### 1. 索引的类型

数据库中用来确定索引顺序的字段，称为关键字字段，也称为索引关键字。在 Visual FoxPro 中，可以建立 4 种类型的索引。

##### 1) 主索引

主索引又称为主关键字，在数据表中只能有一个，它可确保字段中数据的唯一性，同时也决定了数据表中记录的排序依据。在 Visual FoxPro 中，自由表不能建立主索引。

##### 2) 候选索引

候选索引又称为候选关键字，它类似于主索引，但候选索引不仅适用于数据表，也适用于自由表，而且每个表可以建立多个候选索引。

##### 3) 普通索引

普通索引又称为一般关键字或次关键字，它也可以作为排序的依据。普通索引的最大特点是该字段中的数据不一定是唯一的，可以重复，因此排序或查询时会列出所有符合条件的记录。一个数据表中可以建立多个普通索引。

##### 4) 唯一索引

作为唯一索引的字段，称为唯一关键字，它主要为兼容旧版本而设定的。唯一索引用来

选定一组记录，并根据指定字段的值第一次出现的位置来选择和排列记录。实际上，这种功能可以通过建立查询或视图来实现。

## 2. 索引文件

在 Visual FoxPro 中，为数据表建立的索引都存储在以.cdx 为扩展名的复合结构索引文件中，在【项目管理器】窗口中可以查看该索引文件。在打开数据表的同时，索引文件也会随之打开。实际上，在 Visual FoxPro 中，索引文件有两种结构：一种是传统的.idx 索引文件，只有一个索引关键字表达式。另一种是复合索引文件.cdx，包含了多个索引关键字表达式。这些索引关键字表达式称为索引标记。复合索引文件就好像是将多个.idx 文件合成在一个文件中一样。复合索引文件又分为两种：结构化复合索引文件.cdx 和非结构化复合索引文件.cdx。

### 1) 结构化复合索引文件

以结构化复合索引文件存储索引时，索引文件的名称和表名相同，但扩展名为.cdx。只要打开表，就会打开结构化复合索引文件。主索引和候选索引必须用结构化复合索引文件来存储，以确保修改表时可以将索引打开，这是因为主索引和候选索引都需要唯一值，只要改变其值就必须打开。

此外，为了维护参照完整性，也必须将主索引和候选索引与表一起打开。

### 2) 非结构化复合索引文件

非结构化复合索引文件与结构化复合索引文件的不同之处在于：当表打开时，非结构化复合索引文件不会自动打开。通常在需要某一特殊命令而又不想更新索引时，可以使用非结构化复合索引文件。

### 3) 独立.idx文件

独立.idx 文件是只包含一个索引关键字的文件，与非结构化复合索引文件一样，独立.idx 文件必须人工打开。

## 3. 建立索引

建立索引的具体操作步骤如下。

(1) 在【项目管理器】窗口中选择需要建立索引的表，然后单击【修改】按钮，出现【表设计器】对话框。

(2) 在【表设计器】对话框中，单击【索引】选项卡。

(3) 在【索引名】文本框中，输入索引名。在命名索引时，只能使用字母、汉字、下划线和数字，索引名必须以字母或汉字开头，长度不能超过 10 个字符。

(4) 从【类型】下拉列表框中，选定索引类型。

(5) 在【表达式】文本框中，输入作为记录排序依据的字段名。如果只想输出符合条件的记录，可在【筛选】文本框中输入筛选表达式，设定筛选条件。

(6) 单击【确定】按钮即完成索引建立。

以“学生表”为例，可以建立以下两种索引：

- 主索引：在“学生表”的各个字段中，“学号”字段是唯一的，可将其作为主关键字，设定按升序排列。
- 普通索引：“学生表”中的入学成绩不是唯一的，用户往往又要查看入学成绩由低到高排列的顺序，因此将该字段设为普通索引，即次关键字。

本例中的具体操作步骤如下。

(1) 在【项目管理器】窗口中选择“学生表”，单击【修改】按钮，出现【表设计器】窗口。

(2) 在【字段】选项卡中，单击“学号”、“专业编号”字段右边【索引】栏的下拉箭头，选择“升序”排列，如图 3.42 所示。

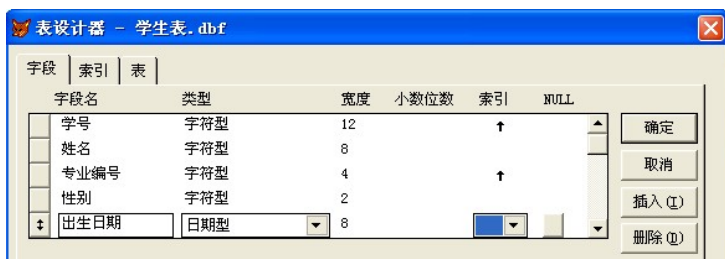


图 3.42 在【字段】选项卡中，对“学号”、“专业编号”字段选择“升序”排列

(3) 单击【索引】选项卡。在该对话框中已有“学号”、“专业编号”字段的索引的相关信息。在“学号”字段的索引类型中选择“主索引”。在“专业编号”字段的索引类型中选择“普通索引”，如图 3.43 所示。

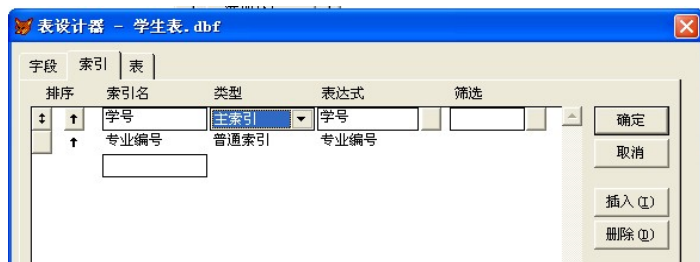


图 3.43 “学号”、“专业编号”字段的索引类型

(4) 至此，两项索引定义结束，单击【确定】按钮，在出现的确认对话框中单击【是】按钮，则索引建立完毕。

可在该数据库的“学生表”中查看索引项，或在【项目管理器】窗口的“学生表”中查看索引项。

#### 4. 修改索引

在【表设计器】窗口中单击【索引】选项卡，同建立方法类似，可以在此修改已建立的索引项，如索引名、类型、排序规则等。同时，也可以利用窗口中的【插入】和【删除】按钮，插入或者删除索引。

同样，如果选择按“学号”进行索引，则记录就会按学号由低到高进行升序排列。同时，

由于按学号建立的是主索引, 根据规定, 作为主关键字的数值必须唯一。如果在“学生表”中输入两个同样的学号, 则系统会拒绝接受, 并提示出错信息。具体操作由用户自行验证, 这里不再赘述。由此看来, 利用主索引不仅可以进行数据的排序, 而且还可以防止数据的重复输入。

### 3.2.2 筛选记录

当对表中的记录进行操作时, 有时不仅要求记录的排列顺序, 而且还需要指定记录的范围、记录的筛选条件。可以在建立索引时加入表达式来筛选所需要的记录。

以刚刚建立的“学生表”的索引为例, 按“入学成绩”来排列数据, 具体操作步骤如下。

- (1) 在【项目管理器】窗口中选择“学生表”, 并打开该表的浏览窗口。
- (2) 单击【表】菜单, 选择【属性】命令, 出现【工作区属性】对话框, 如图 3.44 所示。
- (3) 在【索引顺序】下拉列表框中列出了“学生表”中的所有索引项。其中, “无顺序”表示不按照任何索引项来排列, 只按照原始的输入顺序来显示。
- (4) 选择索引项“入学成绩”来排列数据。单击【确定】按钮, 则记录按照“入学成绩”进行升序排列, 如图 3.45 所示。

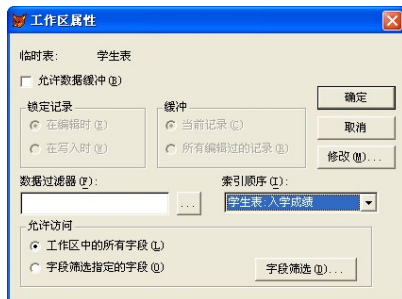


图 3.44 【工作区属性】对话框

学号	姓名	专业编号	性别	出生日期	入学时间	入学成绩	团员否	照片	简历
0641104112	郝明雅	41	女	10/31/83	09/01/08	471	F	Gen	memo
074101067	李巧兰	41	女	05/03/83	09/01/07	473	F	Gen	memo
084205255	李雅迪	42	女	07/08/83	09/01/08	478	T	Gen	memo
084201002	夏利华	42	女	09/04/83	09/01/08	497	F	Gen	memo
084202152	殷 俐	42	女	01/02/83	09/01/08	506	T	Gen	memo
074202123	李林萍	42	女	03/05/84	09/01/07	516	T	Gen	memo
070403128	赵晓飞	04	男	04/06/83	09/01/07	521	F	Gen	memo
080301001	张跃林	03	男	10/23/83	09/01/08	589	T	Gen	memo
080301020	张文斌	03	男	05/20/84	09/01/08	593	T	Gen	memo
080302045	陈江城	03	男	06/04/83	09/01/08	598	T	Gen	memo
070401032	刘慧琴	04	女	07/30/83	09/01/07	599	T	Gen	memo
080401081	孟江浩	04	男	09/20/84	09/01/08	623	T	Gen	memo

图 3.45 记录按照“入学成绩”进行升序排列

## 练习与操作

使用菜单方式给数据库表“教师表.dbf”建立索引。

**练习 1** 对“教师表.dbf”按“教师编号”升序建立主索引, 索引标识为教师编号。

**练习 2** 打开“教师表.dbf”的设计器, 在【字段】选项卡的【索引】栏中选定向上箭头即升序索引(向下箭头即为降序索引)。本练习以“教师编号”升序排序。选定【索引】选项卡, 在该选项卡中可输入关键字表达式, 还可以选择索引类型, 并且【字段】选项卡中建立的索引也能在此选项卡显示出来。【索引】选项卡中【索引名】栏下的空白文本框用来直接输入新索引; 【插入】按钮用于在当前行前插入一个空行, 以供键入新索引; 【删除】按钮则用于删除选定的索引。已建立好的索引如图 3.46 所示。

**练习 3** 对“教师任课表.dbf”按“教师编号”和“课程编号”升序建立普通索引, 索引标识为“教师编号”和“课程编号”。

**练习 4** 对“课程表.dbf”按“课程编号”升序建立主索引, 索引标识为“课程编号”。

**练习 5** 对“学生表.dbf”按“学号”升序建立主索引, 索引标识为“学号”, 按“专业编号”升序建立普通索引, 索引标识为“专业编号”。

**练习6** 对“学生选课表.dbf”按“学号”和“课程编号”升序建立普通索引，索引标识为“学号”和“课程编号”。

**练习7** 对“专业表.dbf”按“专业编号”升序建立主索引，索引标识为“专业编号”。

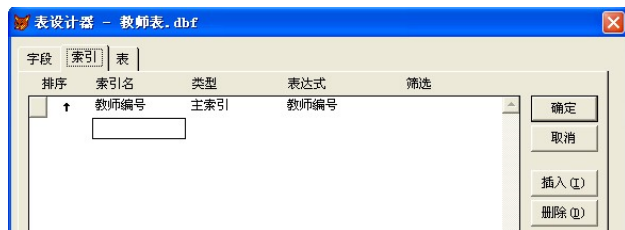


图 3.46 “教师表.dbf”设计器的【索引】选项卡

## 3.3 Visual FoxPro中的命令操作

### 3.3.1 Visual FoxPro命令操作的基本概念

Visual FoxPro 既能与前期的数据库软件产品相兼容，又能适应当前图形用户界面的发展趋势。从 dBASE 到 Visual FoxPro，都可以支持两类不同的基本操作方式，即交互式操作方式和程序操作方式。

#### 1. 交互式操作方式

Visual FoxPro 的交互式操作方式有命令执行与界面操作两种类型。

① 命令执行是传统的交互式操作方式，用户在【命令】窗口输入所需的命令，即可在屏幕上显示执行的结果。这种方式简明快捷，但需要记住命令的格式与功能。

② 界面操作方式则不需要记住命令的格式与功能，易学易用、深受用户欢迎。Visual FoxPro 提供了大量的向导、设计器、生成器等界面操作工具，其设计器普遍配有工具栏，大多数设计器还提供快捷菜单，内含常用的菜单选项，供用户随时调用。

在交互式操作方式下，要完成某项任务均可采用上述两种方法。界面操作方式与直接输入命令这两种方法是一致的。几乎所有的命令功能（但不是全部）都可以通过相应的菜单选择来实现。事实上，当用户选择了某一菜单命令并执行它时，在【命令】窗口中便会自动显示它们对应的键盘命令，就好像用户通过键盘输入了该命令一样，所以在 Visual FoxPro 中，用户可以任意选用或交替使用这两种方法。

命令操作方式简明快捷，而且手工输入各种命令和函数也是学习 Visual FoxPro 的一种手段。对命令语言了解越多，对软件掌握得就越多、越深。而要使用 Visual FoxPro 进行编程，就必须学会这些命令和与函数的用法，因为这些命令和函数就是程序的基础。

#### 2. 命令的一般格式

不同的计算机语言其语言格式不同，就像生活中使用的汉语、英语一样，在书写或会话时，要遵照其各自的语法规则。Visual FoxPro 提供的命令也是如此。为了熟练掌握各种命令的使用方法，首先要了解命令的语法规则。

### 1) 格式

命令的格式如下所示:

```
<命令动词>[<范围>][FIELDS<字段名列表>][FOR<条件>][WHILE<条件>][TO FILE<文件名>/TO  
PRINTER/TO ARRAY<数组名>/TO<内存变量>][ALL[LIKE/EXCEPT<通配符>]][IN<别名/工作区>]
```

### 2) 说明

从上述命令格式可以看见, Visual FoxPro 命令主要由 8 个部分组成, 各部分的功能如下。

① 命令动词。它是 Visual FoxPro 的命令名, 指定计算机要完成的操作。例如, 打开“学生表”的命令为

```
USE 学生表.dbf
```

显示“学生表”中“入学成绩”大于 500 分的学生的姓名、专业的命令为

```
LIST ALL FIELDS 姓名, 专业编号 FOR 入学成绩 >= 500
```

② 范围。范围用来指定命令可以操作的记录集合, 可以有下列 4 种选择。

- ALL: 当前表中的全部记录。
- NEXT <n>: 从当前记录开始的连续  $n$  条记录。
- RECORD <n>: 当前表中的第  $n$  号记录。
- REST: 从当前记录开始到最后一条记录为止的所有记录。

③ 表达式表。表达式表是一个或多个由逗号分隔开的表达式, 用来指示计算机执行该命令所操作的参数。例如, 上述命令的表达式为“姓名, 专业编号”。表达式是由数据和运算符一起构成的有意义的式子。如上面的“姓名”是一个表达式, “专业编号”也是一个表达式, 这些只有数据而没有运算符的单个变量是最简单的表达式。表达式除了可以是字段名、字段名表之外, 还可以加入运算符, 如“职工工资 $\times 1.03$ ”也是一个表达式。

④ FOR<条件>。它规定只对满足条件的记录进行操作, 如果使用 FOR 子句, Visual FoxPro 6.0 将记录指针重新指向表文件项, 并且用 FOR 条件与每条记录进行比较。

⑤ WHILE<条件>。在表文件中, 从当前记录开始, 按记录顺序从上向下处理, 一旦遇到不满足条件的记录, 就停止搜索并结束该命令的执行。在 FOR 子句和 WHILE 子句中, <条件>必须返回逻辑值, 系统会根据逻辑值的真或假来决定操作。

⑥ TO FILE<文件名>/TO PRINTER/TO ARRAY<数组名>/TO<内存变量>。它控制操作结果的输出, 有些命令允许结果向文件输出(利用 TO FILE<文件名>), 有些命令允许操作结果向打印机输出(利用 TO PRINTER), 有些命令允许操作结果向内存变量输出(利用 TO<内存变量>)或向数组输出(利用 TO ARRAY<数组名>)。

⑦ ALL[LIKE/EXCEPT<通配符>]。它指出包括或不包括与通配符相匹配的文件、字段或内存变量。在通配符中可以使用“?”和“\*”。其中“?”和“\*”的使用方法同 MS-DOS 中的用法一致。

⑧ IN<别名/工作区>。它允许在当前工作区操作指定工作区(通过<别名/工作区>指定)中的表文件。

### 3. 命令的输入与编辑

#### 1) 命令的输入

打开 Visual FoxPro 的【命令】窗口，输入需要操作的命令，最后按 Enter 键，就可以执行该命令了。一般一行执行一条命令，如果命令太长则需要续行。

#### 2) 命令的编辑

和其他的文本窗口一样，【命令】窗口也是一个可以编辑的窗口，可以在【命令】窗口中进行各种编辑操作，如插入、删除、复制、剪切等，或者用光标和滚动条在整个【命令】窗口中上下移动。这些特性对命令输入起到了很大的帮助作用。比如要输入一个和上一次命令相似的命令，那么只需将光标移动到上一条命令上，然后输入或删除命令的不同部分，最后按下 Enter 键，就可以执行这条新命令了。这时大家也许会问，这样操作不是修改了上一次的命令了吗？通过具体的实践就会发现原来的命令还存在，只是在【命令】窗口的下面多了刚执行的那条新命令。

在【命令】窗口中进行操作时，只需对第 1 条命令和第 2 条命令进行操作，后面的几条命令只需在这两条命令上加以修改即可。

可见，在 Visual FoxPro 中，用户修改【命令】窗口中的命令时，系统实际上修改的是该命令的一个副本。按下 Enter 键后，Visual FoxPro 系统就会发生响应，复制这个修改后的命令，并放到【命令】窗口中作为一条新的命令执行。

如果要重复执行某一条命令，只需将光标移动到该命令上按下 Enter 键即可。和上面一样，Visual FoxPro 系统也是将该命令的一个副本放到【命令】窗口的下面作为一条新的命令来执行。

### 4. 命令的有关规则

Visual FoxPro 命令有以下基本规则。

① 命令动词必须是命令行的第一个非空字符，各子句可以按照任意顺序跟在动词后面。命令动词与子句、子句与子句之间用一个或多个空格隔开，例如：

```
LIST FIELDS 姓名,专业编号 ALL FOR 入学成绩 <= 500
LIST ALL FIELDS 姓名,专业编号 FOR 入学成绩 <= 500
LIST FOR 入学成绩 <= 500 ALL FIELDS 姓名,专业编号
```

② 命令动词和 Visual FoxPro 保留字一般可用前 4 个或 4 个以上字母的简写形式，例如：

```
DISPLAY MEMORY
DISPL MEMO
DISP MEMO
```

③ 当发现某一条执行命令很长时，虽然【命令】窗口可以上下左右滚动，也可以在一行把命令输完，但屏幕的左右滚动会让人感到很不舒服，编辑操作也很不方便，这时不妨尝试一下续行操作。在 Visual FoxPro 中，命令行的总长度不得超过 2048 个字符（包括空格在内）。若命令较长，在显示器上显示不下时，可以分为几行书写。分号（;）用来表示下一行仍是同一命令的一部分，这样就可以把一条长命令分成好多行来书写。进行续行操作时，应注意以下几点：



- 当一条命令被分成多行输入时，如果想同时使用&&命令加入一些注释，那么此时注释不能出现在分号之后，而只能将注释放在命令的最后一行的后面。
- 当准备执行一个被分成多行的命令时，可将光标放在该命令的任意一行上，然后按下Enter键。
- 如果想在一条被分成多行的命令中插入一个新行时，可以按下Ctrl+Enter组合键。
- 命令的最后一行不能以分号结尾。例如：

```
REPLACE ALL 实发工资 WITH (基本工资 + 职务津贴 + 奖金 - 养老保险) * 1.03;  
FOR 工龄 > 20
```

④ 命令、关键字、变量名和文件名中的字母既可以大写也可以小写，还可以大写、小写混合，三者等效。

⑤ 在表达式中，由FIELDS引导字段名表时，字段名表中的各字段间必须用逗号分隔，但必须是在英文状态下输入的西文逗号“,”。

## 5. 命令书写规则

### 1) 文件命名方法

文件名由主文件名和扩展名两部分组成。主文件名最长由10个字符组成，可以包括字母、数字、下划线、连字符等。但加入到数据库中的表和字段名均支持长文件名。扩展名由“.”加3个字母组成，表示文件类型。

### 2) 本书的命令、函数符号的约定

Visual FoxPro (VFP) 的命令和函数都是由一个或多个不同“成分”组成的，正是这些不同的“成分”决定了每一条VFP命令或每一个VFP函数的特定功能。在书写时，为了便于叙述，常引入下列几个符号，其约定如下。

① “[ ]”：其中的内容是可选项，不选时系统自动取默认值。但在程序输入时或在【命令】窗口中输入时均不书写 “[ ]”，而只写其中参数的内容。

② “<>”：其中的内容是用户的选择项，通常有多种可能供用户选择一种。若“<>”不在 “[ ]” 内，则为必选项，即用户必须选择多种可能中的一种。若“<>”在 “[ ]” 内时，当不选择 “[ ]” 中的内容时，“<>” 中的内容也不能选；而当选择 “[ ]” 中的内容时，其 “[ ]” 中“<>” 内的内容就为必选项。但在程序输入时或在【命令】窗口中输入时均不书写 “[ ]” 和“<>”，而只写其中的参数内容。

③ “/”：为二选一表示符。要求用户从本符号的左右两项中选择一项。同样，在命令或函数的输入中，“/” 也不要写。

④ “...”：省略符。它表示在一个命令或函数表达式中，某一部分可以按同一方式重复。

提示：在实际操作中不会输入上述符号。

## 6. 设置格式

### 1) 设置字体

在进行命令输入的过程中，可以通过改变【命令】窗口中的字体大小、行间距等清晰地显示每一条命令。用户可以使用【格式】菜单中的【字体】选项来改变字体的大小。

## 2) 设置行缩进

行缩进可以极大地改善被分成多行的命令的可读性。在【命令】窗口中为了产生缩进效果，在输入命令前先按下 Tab 键，当然也有可能需要多按几次 Tab 键以产生更多的行缩进，这样，接下来的行就自动产生相同的缩进。一旦在【命令】窗口中加入一个行缩进，那么接下来的行就可自动产生相同的缩进，不过此时需要按 Ctrl+Enter 组合键将光标移到下一行，而不能使用 Enter 键。

## 7. 出错处理

在【命令】窗口中输入命令时，难免会犯一些错误，比如输错了命令或者命令不完整，当按下 Enter 键时，Visual FoxPro 就会给出一个很简单的出错信息，指明用户的错误。例如，将 MODIFY STRUCTURE 命令输成了 MODIDY STRUCTRUE，那么系统会给出这样的提示“命令中含有不能识别的短语或关键字”，提示命令出错。如果输入一条命令不完整，比如建立一个表索引，如果仅仅输入了 INDEX ON 就按 Enter 键，则系统会给出“缺少表达式”提示，告诉你虽然这条命令存在，但是命令中必要的成分未输全。

在 Visual FoxPro 中，错误信息提示窗口是以对话框的形式出现的，对话框的形式比起原来的版本多增加了两个按钮：【确定】和【帮助】按钮。单击【确定】按钮就关闭了错误信息对话框，单击【帮助】按钮可寻求在线帮助，以找到产生问题的源头。

出现错误信息对话框后，按下 Enter 键或单击【确定】按钮均可以将该对话框关闭。一旦出现错误，就要根据相应的提示检查是否输错了命令，或者对照函数手册查看是否命令输入完整。在中文环境下输入时，如果怎样检查一条命令都发现不了错误，但就是不能执行，这就应考虑是否在命令中输入了不可见字符。此时可将原命令重新输入一遍，也许就可以正常运行了。

## 3.3.2 数据库、数据表的操作命令

### 1. 创建一个数据库的命令

格式：CREATE DATABASE [<数据库文件名>|<?>]

功能：创建一个数据库并打开它。“?”是显示【创建】对话框，可在此指定数据库的名称。

示例：CREATE DATABASE 数据库 1

### 2. 打开一个数据库的命令

格式：OPEN DATABASE <数据库文件名>

功能：打开一个数据库。

示例：OPEN DATABASE 数据库 1

### 3. 打开数据表的命令

格式：USE <数据表文件名>

功能：打开指定路径下的数据表。

示例：USE c:\liuli\abc\y2\_1.dbf

#### 4. 关闭数据表的命令

格式: CLOSE 或 CLOSE DATA (CLOSE ALL)

功能: 关闭已打开的数据表。

示例: CLOSE

#### 5. 复制并创建新的数据表文件的命令

格式: COPY TO <库表文件名>

功能: 将当前打开表的全部或部分复制到一个新表中。

示例: COPY TO c:\liuli\abc\s2\_1.dbf

#### 6. 复制表结构并创建新表的命令

格式: COPY STRUCTURE TO <库表文件名>

功能: 将一个已打开的表的全部或部分结构（即字段名）复制成一个新的表。

示例: COPY STRUCUTRE TO c:\liuli\abc\s2\_2.dbf

#### 7. 表结构的显示

格式: LIST|DISPLAY STRUCTURE

功能: 显示当前表结构的有关信息。

示例: LIST STRU

#### 8. 表结构的修改

格式: MODIFY STRUCTURE

功能: 显示表设计器, 修改当前表文件的结构。

示例: MODI STRU

### 3.3.3 表记录的操作命令

#### 1. 记录的编辑修改命令: EDIT、CHANGE、BROWSE

格式: EDIT [<范围>] [FIELDS<字段名表>] [FOR|WHILE<条件>]

CHANGE [<范围>] [FIELDS<字段名表>] [FOR|WHILE<条件>]

BROWSE [<范围>] [FIELDS<字段名表>] [FOR|WHILE<条件>]

功能: 均为全屏幕编辑命令, 用于显示和编辑表中记录。

示例: EDIT FOR 专业编号 = "42" FIEL 学号, 姓名, 性别, 入学成绩, 团员否  
CHAN FOR 性别 = "女" FIEL 姓名, 入学成绩, 专业编号  
BROW FIEL ysf, yf, skrq, bz

#### 2. 更新指定记录的命令: REPLACE

格式: REPLACE [<范围>] [FIELDS<字段名>] [FOR<条件>] WITH <表达式>

功能: 可用一个统一的表达式一次修改一批数据。

示例: REPL REST 月收入 WITH 月收入\*(11+0.15) for 工龄>=20

### 3. 记录的显示命令：LIST、DISPLY

格式：LIST|DISPLY [<范围>] [FIELDS<字段名表>] [FOR<条件>]

功能：显示当前表中全部或部分记录和数据。LIST 为滚屏显示，DISPLY 为分屏显示。

示例：LIST|DISP NEXT 6 FIEL 姓名，系名，授课班级，授课人数，课时

### 4. 查找表中指定记录的命令：LOCATE

格式：LOCATE [<范围>] [FOR|WHILE <条件>]

功能：在指定范围内将记录指针移到满足条件的第一条记录上。如果继续查找，在【命令】窗口输入 CONTINUE 命令。

示例：LOCATE FOR 职称="教授"

CONTINUE

### 5. 移动记录指针命令：GOTO、SKIP

#### 1) 绝对移动命令

格式：GO[TO] <数值表达式> /BOTTOM/TOP

功能：将记录指针不附带任何条件地移动到指定记录号上。

说明：GO 和 GOTO 写法是等效的。数值表达式值的整数部分应当大于或等于 1 且小于或等于当前库文件中的记录总数。当正数值表达式是常量时，GO 或 GOTO 可略去不写，尽管在编程时并不提倡这一省略。

示例：GOTO 20（或 GO 20）

GOTO BOTTOM

#### 2) 相对移动命令

格式：SKIP <数值表达式>

功能：相对移动记录指针是指相对当前记录位置移动记录指针。

说明：数值表达式的值可以是正数、负数或零。设表达式值的整数部分为  $n$ ，则相对当前记录位置将记录指针向前或向后移动  $n$  个记录。具体地说：

- 当  $n < 0$  时，相对当前记录位置记录指针向前移动  $n$  个记录。
- 当  $n = 0$  时，记录指针不变。
- 当  $n > 0$  时，相对当前记录位置记录指针向后移动  $n$  个记录。若  $n$  为 1，可以直接写成 SKIP。

示例：SKIP 5

SKIP -2

SKIP（相当于 SKIP 1）

### 6. 删除记录命令：DELETE、PACK

#### 1) 逻辑删除命令

格式：DELETE [<范围>] [FIELDS<字段名表>] [FOR<条件>]

功能：给符合条件的记录打上删除标记。

示例：DELETE ALL FOR 性别="女"

## 2) 物理删除命令

格式：PACK

功能：将打上删除标记的记录从数据表中彻底删除。

示例：PACK

## 7. 删除表中的全部记录的命令：ZAP

格式：ZAP

功能：删除表中的所有记录。

示例：ZAP

## 8. 恢复已加删除记号的命令：RECALL

格式：RECALL [<范围>] [FIELDS<字段名表>] [FOR<条件>]

功能：取消给记录添加的删除标记。

示例：RECA ALL FOR 性别 ="女"

### 3.3.4 追加记录命令

#### 1. 添加记录的命令

格式：APPEND [BLANK]

功能：在当前表的末尾添加记录。若有 BLANK 选项，系统不进入全屏幕编辑，直接在表的末尾添加一条空记录，其内容以后可以用其他命令填写。

示例：APPE

#### 2. 在表中插入记录

格式：NSERT [BLANK] [BEFORE]

功能：在当前记录之前或之后插入一条或多条新记录。

示例：INSERT BLANK

#### 3. 添加一批记录

格式：APPEND FROM <数据源文件名>/? [FIELDS<数据源字段名列表>];  
[FOR<条件>] [[TYPE]<文件类型>]

功能：用于将一批记录添加到当前表的尾部。记录既可来自其他表，也可来自其他格式的数据文件。

示例：可以用下面的命令将“学生表”中“入学成绩”大于 530 分的记录添加到“学生表 2”中，再将符合要求的文本文件 y2\_2b.txt 全部追加到当前数据表的尾部。

```
USE d:\liuli\abc\学生表 2
```

```
APPEND FROM d:\liuli\abc\学生表 FOR 入学成绩>=530
BROWSE                                && 看看"学生表 2"中有什么记录
APPE FROM d:\liuli\abc\ksml2\y2_2b.txt sdf
USE
```

### 3.3.5 记录的排序和索引

#### 1. 记录的排序

记录排序是把记录按某些字段的值进行排序，重新组织为一个新表。可以用 **SORT TO** 命令进行记录排序。排序命令的格式如下：

```
SORT TO <目标文件名> ON <字段 1> [/A[/D]][/C];
[, <字段 2>][[/A[/D]][/C]...][ASCENDING/DESCENDING];
[<范围>][FOR<条件>][WHILE<条件>];
[FIELDS<字段列表>FIELDS LIKE<字段框架>
/FIELDS EXCEPT<字段框架>][NOOPTIMIZE]
```

① 排序只能按字段的值进行排序，不能按表达式排序。排序字段由 **ON** 子句指定。排序字段可以是多个，即可以进行多重排序，并以排在前面的字段优先。

② 排序可以是降序，也可以是升序。升序由跟在排序字段后面的 **/A** 指定，降序由跟在后面的 **/D** 指定。如果同时指定 **/A**、**/D**，则系统只承认 **/D**。如果不指定，同时又没有 **[ASCENDING/DESCENDING]** 子句，则系统默认为升序。还可以用 **/C** 规定排序时不区分大小写。每个排序字段的顺序可以不一样。

③ **[ASCENDING/DESCENDING]** 的 **ASCENDING** 对所有不带 **/D** 的字段按升序排序，**DESCENDING** 对所有不带 **/A** 的字段按降序排序。如果省略了 **[ASCENDING/DESCENDING]** 子句，则排序顺序默认为升序。

④ 排序可进行筛选和投影，筛选由 **[<范围>]**、**[FOR<条件>]**、**[WHILE<条件>]** 子句实现，投影由 **FIELDS** 子句实现。

⑤ 在投影操作中，**FIELDS LIKE** 子句表示新表中包含与字段框架匹配的所有字段。**FIELDS EXCEPT** 子句表示除了与字段框架匹配的字段之外所有字段都包含在新表中。字段框架支持通配符。

例如，将“学生表”中的所有记录按“性别”的降序和“入学成绩”的升序排序并生成一个新表，具体操作如下：

```
USE d:\liuli\abc\学生表
SORT TO d:\liuli\abc\学生表 1x ON 性别/D, 入学成绩/A
USE d:\liuli\abc\学生表 1x
BROWSE
```

#### 2. 记录的索引

在 Visual FoxPro 中，索引文件用于存储相关数据表文件的索引信息，它指出关于索引表式每一个值的记录在对应库文件中的逻辑顺序和记录位置（记录号），因此也称为库文件中的逻辑排序文件。以下是建立索引的命令：

```
INDEX ON <关键字表达式> TO <索引文件名> [FOR<条件>][UNIQUE]
```

索引操作的功能就是将当前表文件中的数据记录按指定的关键字表达式的值进行逻辑排序,排序结果存于索引文件中,并不改变当前库文件中数据记录的物理位置。索引命令中各参数的说明如下。

① **FOR<条件>**: 为当前库文件中参加索引的数据记录设定条件。若选择此项,则当前库文件中仅满足条件的记录参加逻辑排序,否则当前库文件中的所有记录都参加逻辑排序。

② **UNIQUE**: 选择此项,则当前库文件中仅具有相同索引表达式值的第一个记录参加逻辑排序(在索引文件中关于索引表达式某一个值的记录唯一);否则,当前库文件中所有记录都可能参加逻辑排序。

排序与索引的异同如下。

① 当排序的关键字只有单个字段名时,两条命令的关键字部分写法完全相同,但若排序的关键字为多个字段名时,在 **SORT** 命令中只需将这多个字段名依次列出,而在 **INDEX** 命令中,则要将它们组合成一个字符型表达式。

② 索引排序只能升序排列,如需降序则要自己设计表达式,而用 **SORT** 命令则能方便地对排列的升序、降序进行选择。

③ **SORT** 命令中的 **FIELDS** 和 **FOR** 等可选项可以在排序的同时实现选择和投影操作,而 **INDEX** 命令不具备这些功能。

④ 执行 **SORT** 命令将产生一个新的排序的表文件,而 **INDEX** 命令只产生一个索引文件,不改变原来表文件的记录顺序。

### 3.3.6 记录的查询

#### 1. 查找常量

格式: **FIND** <常量>

功能: 用于查找指定常量的记录。

说明: 这里的常量是指要查找的关键字,可以是数值型和字符型。对于数值型常量,必须直接写出或者通过宏替换函数替换;对于字符型常量,当字符串左部是非空格常量时,是否用引号括起来都是等效的;但是,当字符串左部有空格时,必须用引号括起来。另外,执行此命令时,要求在当前工作区中打开查找关键字的主索引文件,并且查找关键字的数据类型必须与主索引表达式值的类型一致。使用此命令进行查找时,可以通过 **EOF()**或 **FOUND()**函数的值来判断是否找到相匹配的记录。

```
示例: USE d:\liuli\abc\成绩表
      INDEX ON 课程名称 TO XY
      FIND 大学语文
      ?FOUND()
      DISP
      SKIP
      DISP
      USE
```

**提示：**在 Visual FoxPro 中，所谓匹配，对数值型和日期型来说就是相等。对字符型数据有下列两种情况：在 SET EXACT ON 状态下，匹配就是相等；在 SET EXACT OFF 状态下，匹配就是要查找的关键字是索引表达式值的左部子串。

## 2. 查找表达式的值

格式：SEEK <表达式>

功能：用于查找指定表达式的记录。

说明：这里的表达式是查找关键字，表达式的值可以是数值型、字符型和日期型。此命令也要求在当前工作区中打开查找关键字的主索引文件，并且查找关键字的数据类型必须与主索引表达式值的类型一致。

示例：USE d:\liuli\abc\课程表  
INDEX ON 课程名称 TO xy  
SEEK "C 语言程序设计"  
DISP  
SKIP  
DISP  
USE

## 3.3.7 数值统计

将原始数据输入到计算机后，最终要对其进行统计分析，以便获得具有较高价值的统计资料。Visual FoxPro 提供了功能较强的统计命令。

### 1. 记录个数统计

格式：COUNT [<范围>][FOR<条件>][WHILE<条件>][TO<内存变量>]

功能：在指定范围内，统计满足条件的记录个数，并存入指定的内存变量中。

示例：USE d:\liuli\abc\成绩表  
&& 统计当前成绩表中英语成绩大于 90 分的记录个数，并存入 x 内存变量中。  
COUNT FOR 考试成绩>90 TO x

### 2. 求和命令

格式：SUM <表达式表> [范围][条件][TO<内存变量名>][NOOPTIMIZE]

功能：在指定范围内，将符合条件的记录参加表达式计算后分别累加，并可将累加结果存入对应的内存变量中。

示例：USE d:\liuli\abc\成绩表  
&& 累加当前成绩表中数学成绩大于 90 分的记录的字段值，并存入 y 内存变量中。  
SUM 考试成绩 FOR 等级="良" TO y

### 3. 求平均值命令

格式：AVERAGE <表达式表> [范围][条件][TO<内存变量名>][NOOPTIMIZE]

功能：在指定范围内，符合条件的记录参加表达式计算后保存表达式的平均值，并可将结果存入对应的内存变量中。



示例: USE d:\liuli\abc\成绩表

\* 求出表中英语成绩大于等于 90 分记录的平均值, 并存入 z 内存变量中。

```
AVERAGE 考试成绩 FOR 等级 ="良" TO z
```

## 4. 分类求和

格式: TOTAL TO <文件名> ON <字段名> [FIELDS<字段名表>];  
[范围][条件] [NOOPTIMIZE]

功能: 在指定范围内, 将符合条件的记录中数值型字段按指定字段值相同的记录分别累加, 并存入指定的表文件中。

示例: USE d:\liuli\abc\成绩表

&& 在当前表中按学号相同的记录中数值型字段分别累加, 并存入 a1 表文件中。

```
TOTAL TO a1 ON 学号 FOR 考试成绩>=90
```

```
USE a1
```

## 练习与操作

练习 1 对“教师表”按如下要求进行操作, 将操作后的新表结构显示出来。

- (1) 复制“教师表”的结构重命名为“教师表 1”, 将复制后的新表的结构显示出来。

```
USE d:\学生成绩系统\教师表.dbf
COPY STRU TO d:\学生成绩系统\教师表 1.dbf
USE d:\学生成绩系统\教师表 1.dbf
LIST STRU
```

- (2) 将“教师表”复制为“教师表 2”。

```
USE d:\学生成绩系统\教师表.dbf
COPY TO d:\学生成绩系统\教师表 2.dbf
```

- (3) 复制一个仅有教师编号、教师姓名、性别、职称 4 个字段的表结构“教师表 3”。

```
COPY TO d:\学生成绩系统\教师表 3.dbf FIELDS 教师编号, 教师姓名, 性别, 职称
```

- (4) 将“教师表”从第 2 个记录到第 5 个记录中职称为副教授的记录复制到“教师表 4”中。

```
USE d:\学生成绩系统\教师表.dbf
GOTO 2
COPY TO d:\学生成绩系统\教师表 4.dbf FOR 职称="副教授" NEXT 4
```

练习 2 对“教师表 3”按如下进行操作。

- (1) 打开“教师表 3”输入一条新记录, 内容为"js03", "张文", "男", "教授”。

```
USE d:\学生成绩系统\教师表 3.dbf
APPE
```

在打开的“教师表 3”的最后按要求输入: "js03", "张文", "男", "教授”。

- (2) 将“教师表”中所有职称为教授或副教授的记录复制到“教师表 3”中。

```
USE d:\学生成绩系统\教师表 3.dbf
APPE FROM d:\学生成绩系统\教师表.dbf FOR 职称="教授" OR 职称="副教授"
```

- (3) 将表中所有性别为女的记录加上删除标记。

```
DELE ALL FOR 性别="女"
```

- (4) 恢复表中所有性别为女的并带有删除标记的记录。

```
RECA ALL FOR 性别="女"
```

- (5) 将教师编号为“js03”的记录彻底删除。

```
DELE ALL FOR 教师编号="js03"  
PACK
```

**练习 3** 打开“教师表 2”，把其中性别为“女”且职称“讲师”的职称改为“副教授”。

```
USE d:\学生成绩系统\教师表 2.dbf  
REPL ALL 职称 WITH "副教授" FOR 性别="女" AND 职称="讲师"
```

**练习 4** 将以上命令复制到 ml.txt 文件中，并存入“学生成绩系统”文件夹中。

选中【命令】窗口的全部命令，按下 Ctrl+C 组合键。打开记事本程序，单击鼠标右键，选择快捷菜单中的【粘贴】命令。在记事本程序的【文件】菜单中选择【保存】命令，输入文本文件名“ml.txt”，存入指定文件夹“学生成绩系统”中。

**练习 5** 求出“学生表”中女生入学成绩的平均值并显示。

```
USE d:\学生成绩系统\学生表.dbf  
AVERAGE 入学成绩 FOR 性别="女" TO pjcj  
? "该班女生入学平均成绩为: ",pjcj,"分"
```

**练习 6** 统计男生的人数。

```
USE d:\学生成绩系统\学生表.dbf  
COUNT FOR 性别="男" TO ns  
? "全班共有男生: ",ns,"人"
```

**练习 7** 对“教师表 2”先按照性别升序排列，如果性别相同再按工资降序排列，排序后的结果放在表“教师表 x2”中。

```
SORT TO 教师表 2.dbf ON 性别,工资/d
```

## 3.4 多表操作

在数据库系统中，如何方便、高效地进行数据处理、减少数据的冗余度是一个很重要的问题。根据关系数据库的理论，常常把一个复杂的数据库分解成多个不可再分的数据表，这些表不但能够独立使用，也可以根据需要建立数据之间的连接关系（也称为关联关系），从而提高数据的处理速度及利用效率。通常，数据表之间的连接关系可以分为一对一、一对多、多对多三类。

### 3.4.1 数据表的关联

#### 1. 数据表之间的连接关系

##### 1) 一对一的连接关系

所谓一对一的连接关系，是指在数据表 1 中的任何一个数据记录，在数据表 2 中只能有唯一的一个数据记录与之相对应。例如曾经给出的“学生表”和“学生选课表”，一个学生

在一定的阶段只能有一个成绩记录，则这两个表之间的连接关系为一对一的关系。这是表中最简单的关系。

在 Visual FoxPro 中，数据表之间的连接关系是通过索引来创建的，在一对一的关系中，两个数据表必须有一个相同的索引字段作为主关键字字段，通过这个主关键字索引来完成连接。其中一个数据表称为父表，其关键字字段称为原始关键字字段，另一个数据表称为子表，其关键字字段称为外部关键字字段。

## 2) 一对多的连接关系

所谓一对多的连接关系是指数据表 1 中的任何一个数据记录，在数据表 2 中可以有多个数据记录相对应，而数据表 2 中的任何一个数据记录只能与数据表 1 中的唯一一个数据记录相对应。这是表中最常用的关系。

实际上，一对多的连接关系是在两个数据表中选择一个相同的数据字段，父表即“一”方要根据该相同字段建立一个主索引，而子表即“多”方必须根据该相同字段建立一个普通索引或唯一索引。例如“学生表”和“图书借阅表”，一个学生可以借阅多种图书，但一个学生在“学生表”中只能有一个记录，而在“图书借阅表”中可以有多个记录，则“学生表”和“图书借阅表”之间便是一对多的连接关系。

## 3) 多对多的连接关系

所谓多对多的连接关系是指数据表 1 中的任何一个数据记录都可以与数据表 2 中的多个数据记录相对应，而数据表 2 中的任何一个数据记录也可以与数据表 1 中的多个数据记录相对应。这是表中最复杂的关系。

通常在处理多对多的关系时，都把多对多的关系分成两个不同的一对多的关系，即通过一个中介表来建立两者的对应关系。

## 2. 建立数据表之间的关联

数据表之间关联关系的建立非常简单，以“学生表”和“学生选课表”为例，建立二者之间“一对多”的连接关系。操作方法如下。

(1) 首先确定二者之间的相同字段——学号。

(2) 确定父表——“学生表”，确定子表——“学生选课表”。

(3) 确定“学生表”按照“学号”字段建立了主索引，而“学生选课表”按照“学号”字段建立了普通索引，如果这些工作事先没有完成，可在表设计器中添加索引。

(4) 在【数据库设计器】窗口中，选定“学生表”中的主索引字段“学号”，然后按下鼠标左键，将其拖到“学生选课表”中的对应字段“学号”处放开鼠标左键。此时，可以看到它们之间出现一条黑线，表示在两个数据表之间建立起了如图 3.47 所示的一对多的关联关系。

(5) 建立好关联之后，单击关闭按钮关闭【数据库设计器】窗口。

## 3. 编辑数据表之间的关联

在数据表之间建立关联关系后，如果想删除已建立的关联或重新建立其他的关联关系，可以利用 Visual FoxPro 提供的快捷菜单进行操作。操作方法如下。

(1) 在【数据库设计器】窗口中打开相关联的数据表。

(2) 用鼠标左键单击相关联的关联线，则该关联线加粗显示。

(3) 在表的关联线处单击鼠标右键，弹出快捷菜单。选择【编辑关系】命令，出现【编辑关系】对话框，如图 3.48 所示。

(4) 在【表】和【相关表】中选择其他索引建立关联后，单击【确定】按钮即可。

(5) 若要删除已建立的关联，只需选择快捷菜单中的【删除关系】命令，则关联线消失，表示两数据表之间的关联关系已被删除了。

另外，当用鼠标左键单击两数据表之间的关联线后，还可以选择【数据库】菜单中的【编辑关系】命令来进行相应的操作。

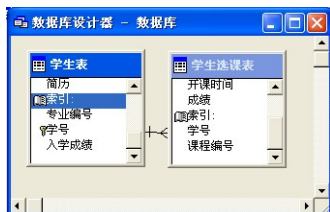


图 3.47 一对多的关联关系

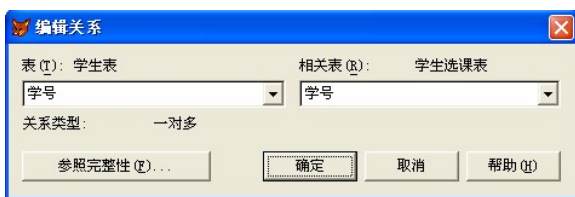


图 3.48 【编辑关系】对话框

## 3.4.2 设置参照完整性

在 Visual FoxPro 中，系统提供了数据表的参照完整性功能，用于确定数据库中表间关系正确的一组规则。在数据表之间建立关联关系后，可以通过设置参照完整性来建立这些规则，以便控制相关表中记录的插入、更新或删除操作。

### 1. 规则说明

#### 1) 更新规则

更新规则用于设置关系数据表之间的更新操作，具体功能包括：

- 级联：用新的关键字值更新子表中的所有相关记录。
- 限制：若子表中有相关记录则禁止更新。
- 忽略：允许更新，不管子表中的相关记录，即不进行任何参照完整性的检查工作，这是默认设置。

如果选择了“级联”设置，则在“学生表”中某一数据记录的学号被更新时，在“学生选课表”中对应记录的学号也将被更新。

#### 2) 删除规则

删除规则用来设置关系数据表之间的删除操作，具体功能也包括：

- 级联：删除子表中的所有相关记录。
- 限制：若子表中有相关记录则禁止删除。
- 忽略：允许删除，不管子表中的相关记录。

如果选择了“限制”设置，当在“学生表”中删除学号为“084202152”的记录时，因为在子表“学生选课表”中存在相对应的记录，则该操作被禁止。

### 3) 插入规则

插入规则用来设置关联数据表间的插入操作，只有如下两种选择：

- 限制：若父表中不存在匹配的关键字值，则禁止插入。
- 忽略：允许插入。

此时，如果选择了“忽略”设置，则在“学生选课表”中添加任何新记录时，即使添加的学号在“学生表”中找不到对应值，也不会有任何限制。

## 2. 设置参照完整性的步骤

仍以“学生表”和“学生选课表”之间的关联为例，下面讲述设置参照完整性的操作方法。

(1) 打开【参照完整性生成器】对话框，有以下几种方式：

- 在【编辑关系】对话框中单击【参照完整性】按钮，即可打开【参照完整性生成器】对话框。
- 右击关联数据表的关联线，在弹出的快捷菜单中选择【编辑参照完整性】命令，即可打开【参照完整性生成器】对话框。选择【数据库】菜单中的【编辑参照完整性】命令，也可打开【参照完整性生成器】对话框。该对话框列出了此数据库中彼此有关联关系的数据表，并指明谁是父表、谁是子表，以及用以建立关联的索引字段，如图 3.49 所示。

(2) 在【参照完整性生成器】对话框中，有【更新规则】、【删除规则】、【插入规则】三个选项卡供用户选择。

(3) 设置完参照完整性之后，单击【确定】按钮。

(4) 在随后弹出的对话框中单击【是】按钮，参照完整性设置便完成了。此后，相关联的数据表之间将根据这种规则自动进行检查和维护，在对数据进行更新、删除和插入时就不会产生数据不对应的情况了。

如果要修改参照完整性，可以在【数据库设计器】窗口中用鼠标右键单击数据表之间的关联线，在出现的快捷菜单中选择【编辑参照完整性】命令，或在【数据库】菜单中选择【编辑参照完整性】命令。在出现的【参照完整性生成器】对话框中，重新设置已有的参照完整性即可。

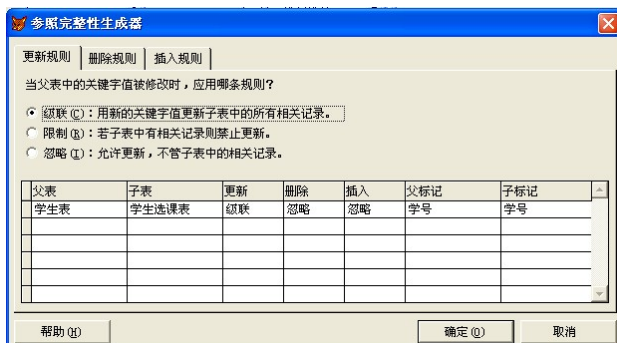


图 3.49 【参照完整性生成器】对话框

## 练习与操作

**练习 1** 在数据库中，利用菜单方式建立各表之间的关联。参考步骤如下。

- (1) 打开“学生成绩系统”的项目管理器，选择【数据】选项卡下数据库中的表，单击【添加】按钮，将所建立的数据表依次添加到数据库中。
- (2) 选择数据库后单击【修改】按钮，随后打开【数据库设计器】窗口，由于各表已经建立了索引，在父表已建立好的索引字段处，按住鼠标左键，将它拖到子表的相应索引处放开，如图 3.50 所示。
- (3) 若要编辑表间关系，在表间关系的关联线上单击鼠标，连线变粗。再在关联线上单击鼠标右键，在快捷菜单中选择【编辑关系】命令。然后在弹出的如图 3.51 所示的对话框中进行关系编辑。

**练习 2** 编辑参照完整性，参考步骤如下。

- (1) 打开【数据库设计器】窗口，在已建关系的表间关联线上，单击鼠标右键，弹出快捷菜单，选择【编辑参照完整性】命令。

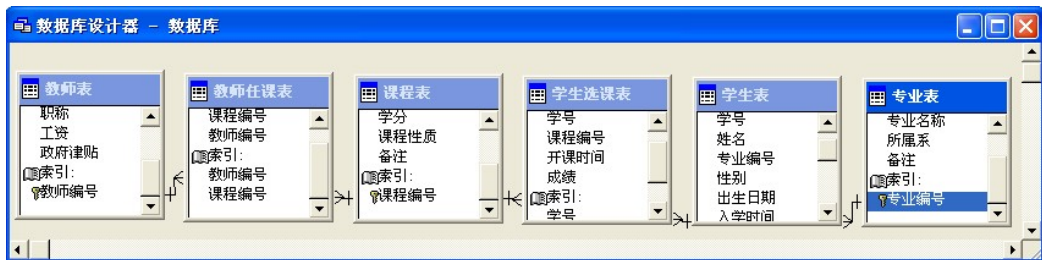


图 3.50 【数据库设计器】窗口

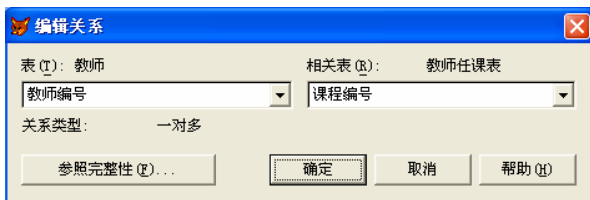


图 3.51 【编辑关系】对话框

- (2) 通过单击【更新规则】、【删除规则】、【插入规则】选项卡来确定对哪个规则进行编辑。在每个选项卡的下方列出了三种选项（级联、限制、忽略），根据需要选择后单击【确定】按钮返回。
- (3) 系统弹出【参照完整性生成器】对话框，提示“是否保存改变，生成参照完整性代码并退出”，单击【是】按钮。
- (4) 系统将继续弹出【参照完整性生成器】对话框，再次单击【是】按钮。系统在执行一系列操作后，返回到【数据库设计器】窗口，此时参照完整性代码生成工作宣告完成。

在设置参照完整性规则时，需要分别设置更新规则、删除规则和插入规则。在实施参照完整性时，这 3 种规则的功能是不一样的。更新规则规定在子表中有相关记录时是否允许更新父表中的关键字值；删除规则规定在子表中存在相关记录时是否允许删除父表中的记录；插入规则规定当父表中没有与之匹配的关键字值时是否允许在子表中插入一个新记录或更新一个已有的记录。

## 思考与练习 3

### 一、思考题

1. 数据库与数据表的关系是什么?
2. 如何删除与恢复记录?
3. 为什么要使用关系数据库?
4. 关系数据库是如何构成的?
5. 什么是一对多关系?

### 二、选择题

1. 若要控制数据库表中的“学号”字段只能输入数字,则应设置\_\_\_\_。  
A. 显示格式      B. 输入掩码      C. 字段有效性      D. 记录有效性
2. 在下列命令中,不具有修改记录功能的是\_\_\_\_。  
A. EDIT      B. REPLACE      C. BROWSE      D. MODI STRU
3. 显示表中所有教授和副教授记录的命令是\_\_\_\_。  
A. LIST FOR 职称 = "教授" AND 职称="副教授"  
B. LIST FOR 职称 >= "副教授"  
C. LIST FOR 职称 = "教授" OR "副教授"  
D. LIST FOR "教授" \$ 职称
4. 打开一个建立了结构复合索引的数据表,表记录的顺序将按\_\_\_\_。  
A. 第一个索引标识      B. 最后一个索引标识  
C. 主索引标识      D. 原顺序
5. 在浏览窗口打开的情况下,若要向当前表中连续添加多条记录应使用\_\_\_\_。  
A. 【显示】菜单中的【追加方式】命令      B. 【表】菜单中的【追加新记录】命令  
C. 【表】菜单中的【追加记录】命令      D. Ctrl+Y 组合键
6. 表文件中有数学、英语、计算机和总分4个数值型字段,要将当前记录的三科成绩汇总后存入总分字段中,应使用命令\_\_\_\_。  
A. TOTAL 数学 + 英语 + 计算机 TO 总分  
B. REPLACE 总分 WITH 数学 + 英语 + 计算机  
C. SUM 数学 + 英语 + 计算机 TO 总分  
D. REPLACE ALL 数学 + 英语 + 计算机 WITH 总分
7. 在【命令】窗口中,显示当前数据表中所有40岁(含40岁)以下,职称为“教授”、“副教授”的姓名和工资,应使用命令\_\_\_\_。  
A. LIST FIEL 姓名,工资 FOR 年龄<=40 AND 职称="教授" AND 职称="副教授"  
B. LIST FIEL 姓名,工资 FOR 年龄<=40 OR 职称="教授" OR 职称="副教授"  
C. LIST FIEL 姓名,工资 FOR 年龄<=40 AND (职称="教授" OR 职称="副教授")  
D. LIST FIEL 姓名,工资 FOR 年龄<=40 OR (职称="教授" AND 职称="副教授")
8. 要求表文件某数值型字段的整数是4位、小数是2位,其值可能为负数,该字段的宽度应定义为\_\_\_\_。  
A. 8位      B. 7位      C. 6位      D. 4位





## 第 4 章 查询与视图

数据库组织并存储信息的主要用途之一是供用户日后查询。查询是数据库管理系统的一项重要功能。用户可以根据需要建立查询，在数据库中检索一条或多条满足条件的记录，供用户进行查看、分析或报表打印。

### 4.1 查询

在 Visual FoxPro 中，设计一个查询主要是通过指定数据源、选择所需字段、设置筛选条件、指定排序依据等工作来完成的。当用户建立好查询后，每次运行查询时，那些符合条件的记录字段信息显示在屏幕上，并且还可以对查询的结果进行排序、分类，作为表单或报表的数据来源等。

Visual FoxPro 系统为建立查询提供了两种方法：一种是查询向导，另一种是查询设计器。可以通过 Visual FoxPro 提供的查询向导和查询设计器来设计查询，也可以使用 SELECT-SQL 语句自己编写查询程序。在查询设计器中设计查询时，它会自动生成对应的 SQL 语句。

#### 4.1.1 查询文件的建立与修改

##### 1. 利用查询向导设计查询

查询向导可以引导用户快速设计一个查询。下面将示例使用查询向导设计一个简单的单表查询。

例如，从“学生成绩系统”数据库的教师表中查询职称为“副教授”的所有男教师的信息，操作过程如下。

##### 1) 启动查询向导

可以有两种方式启动查询向导，如下所示。

① 从系统菜单的【文件】菜单中，单击【新建】命令，在【新建】对话框中选择【文件类型】中的【查询】，然后单击【向导】按钮，启动查询向导。

② 在【项目管理器】窗口中，选择【数据】选项卡中的【查询】，单击【新建】按钮，则出现设计查询方式的【新建查询】对话框，如图 4.1 所示。单击【查询向导】按钮，打开如图 4.2 所示的【向导选取】对话框。

系统提供了三种查询方式，可以分别设计三种不同的查询：

- 查询向导：设计一个一般的标准查询，这是常用的方法。
- 交叉表向导：设计有行和列的交叉查询。
- 图形向导：在 Microsoft Graph 中创建显示 Visual FoxPro 表数据的图形。

在所列出的 3 种向导中, 选择【查询向导】, 再单击【确定】按钮, 进入查询向导的步骤 1—字段选取 (如图 4.3 所示)。



图 4.1 【新建查询】对话框

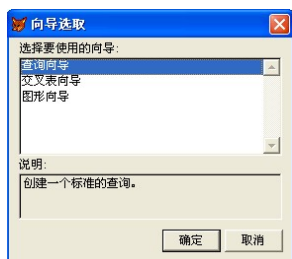


图 4.2 【向导选取】对话框

## 2) 字段选取

① 选择数据库和表。在选择字段之前应先选择查询所使用的数据库和数据表。如果所列表中没有需要的数据库或数据表, 可单击图 4.3 中【数据库和表】列表框右侧的浏览按钮, 从打开的对话框中选择所需要的数据库或数据表。

② 选择字段。例如, 选择“学生成绩系统”数据库的“教师表”, 图 4.3 的【可用字段】列表框中显示出它的全部字段, 这里将全部字段选取到【选定字段】列表框中, 其中单箭头是单个字段选取按钮, 双箭头是全选字段按钮。字段选取结束后单击【下一步】按钮, 进入查询向导的步骤 2—关联表。

## 3) 关联表

在以多个表 (或视图) 作为查询对象的情况下, 必须通过选择相匹配的字段来建立表 (或视图) 之间的联系, 以产生查询的连接条件。比如选择“教师表”中的“教师编号”字段和“教师任课表”中的“教师编号”字段, 作为两表之间相匹配的字段, 单击【添加】按钮, 显示出此关联表达式。单击【下一步】按钮, 进入查询向导的步骤 3—筛选记录。如果创建的是单表查询, 系统会跳过步骤 2—关联表, 直接进入步骤 3—筛选记录, 如图 4.4 所示。



图 4.3 步骤 1—字段选取



图 4.4 步骤 3—筛选记录

## 4) 筛选记录

在步骤 3 中输入查询记录的条件, 本例题中的条件是职称为“副教授”并且性别为

“男”。在【字段】组合框中选择“教师表.职称”字段，在【操作符】组合框中选择“等于”，在【值】文本框中输入“副教授”。选中运算符【与】后，设置第二个条件，在【字段】组合框中选择“教师表.性别”字段，在【操作符】组合框中选择“等于”，在【值】文本框中输入“男”。条件输入结束后，单击【下一步】按钮，进入查询向导的步骤4—排序记录，如图4.5所示。

### 5) 排序记录

查询向导步骤4用于选择排序字段。字段可按升序或降序排序，它将直接影响到查询结果中记录的排列次序。最多可选择三个排序字段，只有当第一个排序字段的值相同时，再按第二个字段的升序或降序排序，依次类推。本例中按“教师编号”升序排序。单击【下一步】按钮，进入查询向导的步骤4a—限制记录，如图4.6所示。

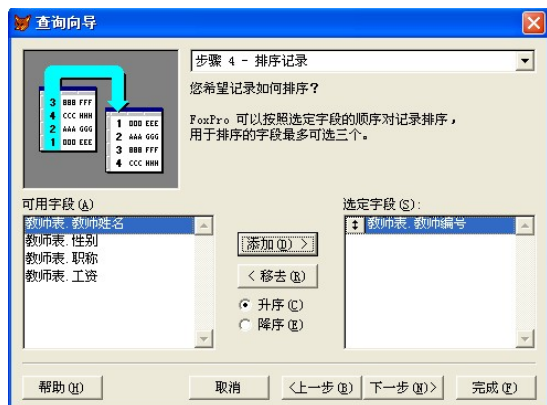


图 4.5 步骤4—排序记录

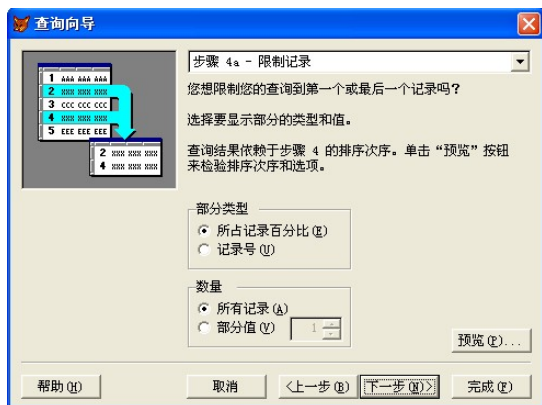


图 4.6 步骤4a—限制记录

### 6) 限制记录

确定查询结果中所包含的记录个数。如果显示符合条件的全部记录，可选择【数量】区域中的【所有记录】；如果只显示符合条件的部分记录，可以使用百分比或记录数来确定要显示的部分记录。限制记录结束后，单击【下一步】按钮，进入查询向导的步骤5—完成，如图4.7所示。

### 7) 完成

这一步用来选择保存后的执行方式，也就是说，选择保存查询设置的类型。这里选择【保存查询】，单击【完成】按钮，出现【另存为】对话框。在【文件名】文本框中输入查询文件名，本例的查询文件名为“教师职称查询”，系统默认的扩展名为.qpr，然后单击【保存】按钮，保存刚刚创建的查询。

按上述操作步骤完成后，在【项目管理器】窗口中，单击【查询】前面的“+”，可以看到其中的查询文件名“教师职称查询”，如图4.8所示。

至此，一个查询就建立好了。在以上操作过程中，要修改前一步所完成的操作，可以单击【上一步】按钮进行修改。整个查询设计过程不能使用查询向导进行修改，但可以使用查询设计器进行修改。查询运行结果如图4.9所示。



图 4.7 步骤 5-完成

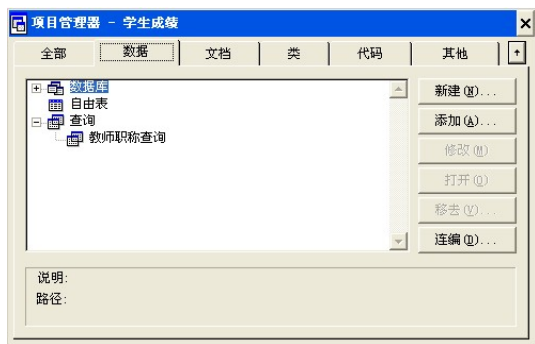


图 4.8 【项目管理器】窗口中的新建查询

教师编号	教师姓名	性别	职称	毕业学校	课程编号	工资	政府津贴
CS0202	赵华林	男	副教授	厦门大学	CS02	2786.00	F
JS0101	桑健中	男	副教授	浙江大学	CS01	3120.00	T

图 4.9 查询运行结果

## 2. 利用查询设计器设计查询

使用查询设计器可以建立任意类型的查询，既包括简单条件的查询，又包括复杂条件的查询；既可以创建计算字段，又可以设置查询结果的输出去向。

### 1) 启动查询设计器

可以有两种方式启动查询设计器，如下所示。

① 选择【文件】菜单中的【新建】命令，在【新建】对话框中选择【文件类型】中的【查询】，然后单击【新建查询】按钮，启动查询设计器。

② 在【项目管理器】窗口中，选择【数据】选项卡中的【查询】，单击【新建】按钮，则出现【新建查询】对话框。单击【新建查询】按钮，则打开【查询设计器】窗口，如图 4.10 所示。



图 4.10 【查询设计器】窗口

系统同时还会出现【添加表或视图】对话框,如图4.11所示。下面用一个具体实例来完成查询的创建。

例如,在“教师表”、“课程表”这两个表中,查询职称是“教授”、“副教授”的教师的情况,并且显示“教师表.教师编号”、“教师表.教师姓名”、“教师表.职称”、“教师表.工资”、“教师表.课程编号”、“课程表.课程名称”、“课程表.学时”等字段的内容。

在【添加表或视图】对话框中,选择数据库中的“教师表”、“教师任课表”、“课程表”,单击【添加】按钮,将3个表添加到【查询设计器】窗口中,并单击【关闭】按钮保存。



图 4.11 【添加表或视图】对话框

## 2) 选择输出字段

字段选项用于选择查询结果中的输出字段。把“教师表.教师编号”、“教师表.教师姓名”、“教师表.职称”、“教师表.工资”、“教师表.课程编号”、“课程表.课程名称”、“课程表.学时”字段添加到【选定字段】列表框中。选定字段结束后,单击工具栏上的运行按钮,可以浏览查询结果,如图4.12所示。

## 3) 关联表

前面已将“教师表”、“教师任课表”、和“课程表”添加到【查询设计器】窗口中,并且它们之间有一条连线,这条连线定义了两个表之间的一种表间连接关系。在添加表时,如果两个表之间没有建立连接关系,则需要两表之间的关联。在关闭【添加表或视图】对话框后,将出现【连接条件】对话框,如图4.13所示。

教师编号	教师姓名	职称	课程编号	课程名称	学时
JC0101	陈得江	教授	CC01	VB.NET程序设计	60
JC0202	赵惠兰	副教授	CC01	VB.NET程序设计	60
JC0303	刘大林	讲师	CC02	数据库原理与应用	60
JS0101	秦健中	副教授	CS01	VFP数据库程序设计	75
JS0202	吴谨敏	副教授	CS01	VFP数据库程序设计	75
CC0101	赵晓兰	教授	CS02	大学计算机应用基础	30
CC0202	李月敏	讲师	CS02	大学计算机应用基础	30
CC0303	王克强	讲师	CC03	多媒体计算机技术	45
CC0404	刘雪燕	副教授	CS01	VFP数据库程序设计	75
CS0101	林星明	教授	CS02	大学计算机应用基础	30
CS0202	赵华林	副教授	CS02	大学计算机应用基础	30
CS0303	季严冬	教授	CC03	多媒体计算机技术	45

图 4.12 浏览查询结果



图 4.13 【连接条件】对话框

在【连接条件】对话框中,左边的组合框称为左字段列表,右边的组合框称为右字段列表。Visual FoxPro 把表间的连接分为4种类型:内部连接、左连接、右连接和完全连接。

① 内部连接: 指在查询结果中,只列出左字段列表与右字段列表相匹配的记录,这是默认设置。

② 左连接: 指在查询结果中,列出左字段列表中的所有记录与右字段列表中相匹配的记录。

③ 右连接: 指在查询结果中,列出右字段列表中的所有记录与左字段列表中相匹配的记录。

④ 完全连接: 指在查询结果中,列出两个字段列表中的所有记录。

根据例题的要求,“教师表”和“教师任课表”是通过“教师编号”建立起连接的,“课程表”和“教师任课表”是通过“课程编号”建立起连接的。在左字段列表中选“教师表.教师编号”字段,在右字段列表中选“教师任课表.教师编号”字段,再选择连接类型(如内部连接),然后关闭【连接条件】对话框。两个表中的编号字段之间出现了一条连线,表明两个表通过编号字段建立了一种连接关系。同样还可以建立“课程表”和“教师任课表”两个表的连接关系。

#### 4) 筛选记录

通过设置筛选条件,将筛选出表中满足条件的记录并显示在查询结果中。方法是在【查询设计器】窗口中,单击【筛选】选项卡。将逻辑表达式“教师表.职称 In "教授","副教授"”输入到所给的输入框中,如图 4.14 所示。

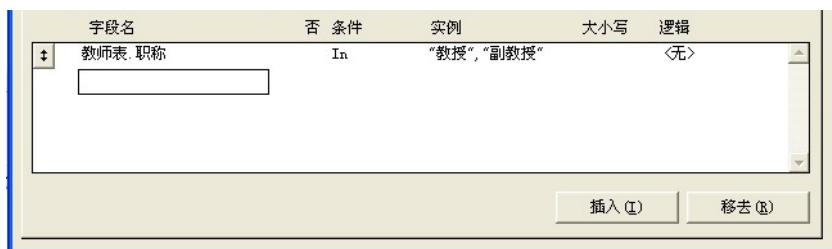


图 4.14 设置查询条件

#### 5) 排序记录

排序查询结果是指按某一个或多个字段的值进行升序或降序显示,例如按“教师编号”从低到高显示查询结果。可在【查询设计器】窗口中,单击【排序依据】选项卡。在【选定字段】列表框中选出用于排序的字段,这里选择“教师表.教师编号”,在【排序选项】中选择【升序】,然后单击【添加】按钮,将其添加到【排序条件】列表框中,如图 4.15 所示。从结果中可以看到记录显示次序是按“教师编号”升序排列输出的。



图 4.15 按“教师编号”升序排列输出

#### 6) 分组查询

分组是指将数据表中某关键字段相同的记录分组生成一条记录。例如在查询结果中,按性别进行分组查询。

在【查询设计器】窗口中,单击【分组依据】选项卡。在【可用字段】列表框中选出用于分组的字段,如“教师表.性别”,并将其添加到【分组字段】列表框中,本例不做选择。



另外，分组就是将一组类似的记录压缩成一个结果记录，这样就可完成基于一组记录的计算。分组在与某些累计函数联合使用时效果最好，例如 SUM()、COUNT()、AVG()等。在计算表达式中可以使用下列函数：

- COUNT(): 求某字段值的指定个数
- SUM(): 求某数值字段的和
- AVG(): 求某数值字段的平均值
- MIN(): 求某数值字段中的最小值
- MAX(): 求某数值字段中的最大值

在每个函数括号中，要求给出一个字段表达式，例如“AVG(成绩.平均分)”。

单击工具栏上的运行按钮，即可浏览查询结果，如图 4.16 所示。关键字字段相同的记录只显示一条。



教师编号	教师姓名	职称	课程编号	课程名称	学时
CC0101	赵晓兰	教授	CS02	大学计算机应用基础	30
CC0404	刘雪燕	副教授	CS01	VFP数据库程序设计	75
CS0101	林星明	教授	CS02	大学计算机应用基础	30
CS0202	赵华林	副教授	CS02	大学计算机应用基础	30
CS0303	季严冬	教授	CC03	多媒体计算机技术	45
JC0101	陈得江	教授	CC01	VB.NET程序设计	60
JC0202	赵惠兰	副教授	CC01	VB.NET程序设计	60
JS0101	秦健中	副教授	CS01	VFP数据库程序设计	75
JS0202	吴莲敏	副教授	CS01	VFP数据库程序设计	75

图 4.16 查询结果

## 4.1.2 查询文件的执行

### 1. 运行查询

运行查询的方法有很多种，例如：

① 在【项目管理器】窗口中，选择【数据】选项卡，单击【查询】前面的“+”，选择要运行的查询文件的文件名，再单击运行按钮，这时查询结果显示在屏幕上。

② 在【程序】菜单中，选择【运行】命令，打开【运行】对话框，选定一个要运行的查询文件后，单击【运行】按钮。

③ 在没退出【查询设计器】窗口之前，单击【查询】菜单中的【运行查询】命令。

④ 在没退出【查询设计器】窗口之前，直接单击运行按钮。

⑤ 在【查询设计器】窗口内单击鼠标右键，在弹出的快捷菜单中选择【运行查询】命令。

⑥ 在【命令】窗口中输入“DO 查询名.qpr”。

### 2. 查询菜单

在【查询设计器】窗口打开后，系统菜单中就会自动增加一个【查询】菜单。该菜单包含【查询设计器】窗口下部窗格中各个选项卡的所有选项，也包含快捷菜单和查询设计器工具栏的大部分功能。下面介绍几条常用命令。

## 1) 查看SQL

SQL 是结构化查询语言 (Structured Query Language) 的缩写, 一条 SQL 语句就可以代替许多条 Visual FoxPro 命令。系统会根据设计条件自动使用 SQL 语言生成相应的语句, 可以通过系统菜单的【查询】→【查看 SQL】命令来查看, 显示由查询操作产生的 SELECT-SQL 命令。显示出来的命令只供阅读, 不能编辑, 但可通过剪贴板进行复制和粘贴, 如图 4.17 所示。

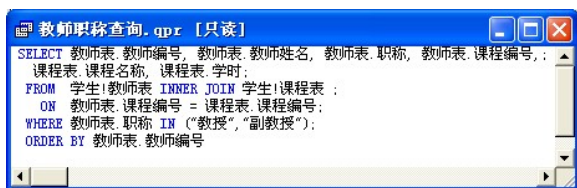


图 4.17 教师职称查询的 SQL 代码

## 2) 查询去向

选择系统的【查询】菜单的【查询去向】命令, 可打开如图 4.18 所示的【查询去向】对话框, 其中共包括 7 个按钮, 表示查询结果的不同输出类型。



图 4.18 【查询去向】对话框

- 【浏览】按钮: 在浏览窗口中显示查询结果, 是默认值。
- 【临时表】按钮: 将查询结果保存于临时表中, 系统不会保存。
- 【表】按钮, 将查询结果作为表文件保存起来。
- 【图形】按钮: 使查询结果以图形的形式输出。
- 【屏幕】按钮: 在当前输出窗口中显示查询结果, 也可指定输出到打印机或文件。
- 【报表】按钮: 向报表文件发送查询结果。
- 【标签】按钮: 将查询结果输出到一个标签文件中。

例如, 可以对“成绩表”文件中每人的总分字段进行数据分析, 从而产生一个分析图, 参考步骤如下。

(1) 在【查询设计器】窗口中添加“成绩表”, 并选定全部字段。

(2) 单击【查询】菜单的【查询去向】命令, 并选择对话框中的【图形】按钮, 再单击【确定】按钮。

(3) 单击工具栏中的运行按钮, 出现【图形向导】对话框, 如图 4.19 所示, 选定字段后单击【下一步】按钮, 进入图形向导的步骤2-定义布局。在【数据系列】列表框添加“成绩”字段, 在坐标轴中添加“姓名”字段, 如图 4.20 所示。





图 4.19 步骤 1—字段选取



图 4.20 步骤 2—定义布局

- (1) 单击【下一步】按钮，出现图形样式的对话框，如图 4.21 所示，这里选定平面柱状图。
- (2) 单击【下一步】按钮，在完成的【图形向导】对话框中，输入一个图形标题，如图 4.22 所示。然后进行预览，如图 4.23 所示。
- (3) 保存为“成绩图表”，如图 4.24 所示。

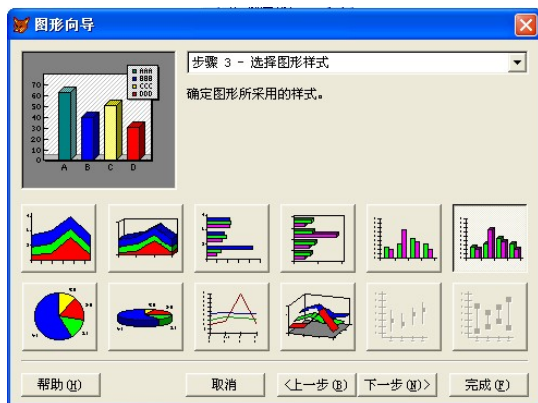


图 4.21 步骤 3—选择图形样式



图 4.22 步骤 4—完成

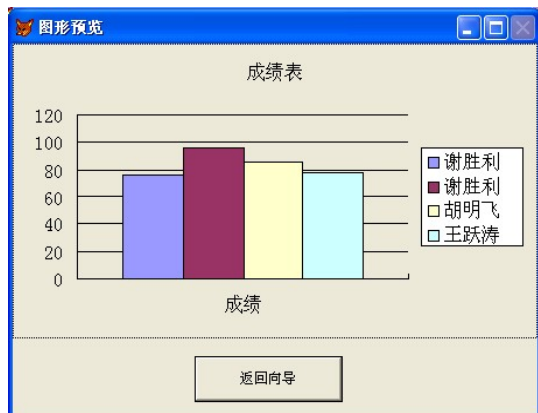


图 4.23 生成柱状图

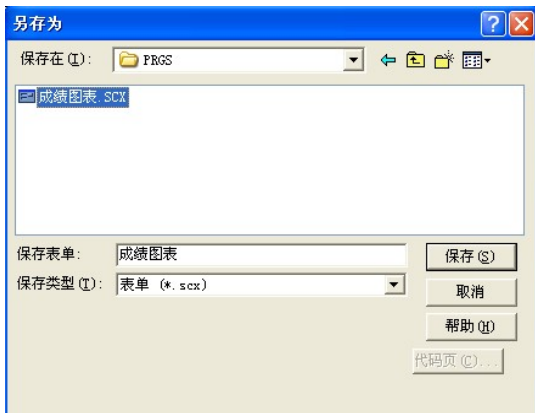


图 4.24 【另存为】对话框

至此，一个完整的查询已经建立起来了，关闭查询设计器，保存创建好的查询，以备随时调用。

### 3) 建立交叉表

在查询中可以利用创建计算字段的功能，对各字段进行求和、求平均值等。在浏览时，把计算字段也显示在一列中。不过有时人们也想把数据表中的某一字段值按行输出，将另一字段值按列输出，在它们的交叉处输出第三个字段的计算值（如总和、平均数、计数、最大值及最小值等）。

例如，利用“成绩表”建立一个交叉表，统计每个学生的各科考试情况，参考步骤如下。

(1) 单击【向导选取】对话框中的【交叉表向导】，打开【交叉表向导】对话框。在【交叉表向导】对话框中选取数据表和可用字段，这里选取“成绩表”的“姓名”、“课程名称”和“成绩”字段，如图 4.25 所示。

(2) 单击【下一步】按钮进入布局对话框。这里把“姓名”字段拖到“行”框中，把“课程名称”字段拖到“列”框中，把“成绩”字段拖到“数据”框中，如图 4.26 所示。

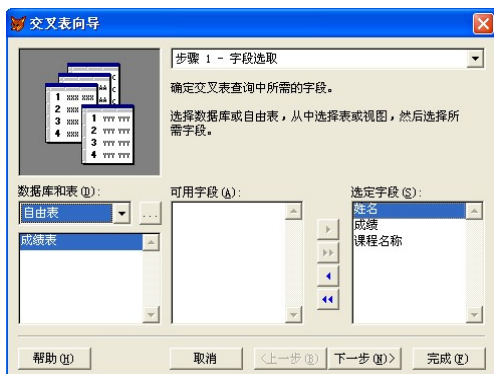


图 4.25 步骤 1—字段选取



图 4.26 步骤 2—定义布局

(3) 单击【下一步】按钮，进入总结信息对话框，如图 4.27 所示。在总结信息对话框中，要求对交叉表中交叉处的数据进行运算，并把运算结果放到交叉表中。

(4) 单击【下一步】按钮进入完成对话框，如图 4.28 所示。单击【预览】按钮，可以预览交叉表查询结果，如图 4.29 所示。单击【完成】按钮，保存交叉表。



图 4.27 步骤 3—加入总结信息



图 4.28 步骤 4—完成

姓名	Vb_net程序	Vfp数据库	大学计算机	多媒体计算	数据库原理	总和
陈江城		72	87			159
郝明雅			67		50	117
李林萍				90		90
李巧兰	87			82		149
李雅迪					91	91
刘慧琴		91			92	183
孟江浩					65	65
夏利华				89	55	144
殷 俐		78	75	85		238
张文斌			88		87	175
张跃林	96	76		87		259
赵晓飞	86					86

图 4.29 预览交叉表结果

在使用向导建立交叉表时，它要求所使用的字段必须属于同一个表或视图，如果这些字段不在同一个表或视图中，则要先建立一个表或视图，并把查询结果输出到一个表中。

## 练习与操作

**练习 1** 利用查询向导创建查询，内容包含“学生表.dbf”中所有学生的学号、姓名及入学成绩，将该查询结果保存为一个名为“R1.qpr”的查询文件，并运行查询文件“R1.qpr”。参考步骤如下。

- (1) 执行【文件】→【新建】命令，在系统弹出的【新建】对话框的【文件类型】区域中选择【查询】单选项，然后单击【向导】按钮。
- (2) 在弹出的【向导选取】对话框中，选择【查询向导】，然后单击【确定】按钮，弹出【查询向导】对话框。
- (3) 在向导的第 1 步中，单击【数据库和表】区域中的浏览按钮，并在打开的对话框中选择“学生表.dbf”所在的存储路径，找到并将其打开。
- (4) 仍在向导的第 1 步中，双击【可用字段】列表框中的“学号”、“姓名”及“入学成绩”字段，使其显示在右侧的【选定字段】列表框中。
- (5) 单击【完成】按钮，弹出查询向导的第 5 步。
- (6) 再次单击【完成】按钮，弹出【另存为】对话框。在其中选择适当的存储路径，并在【文件名】文本框中输入“R1”，然后单击【保存】按钮。
- (7) 执行【程序】→【运行】命令，弹出系统的【运行】对话框。
- (8) 在对话框的【搜寻】列表框中选择所在的存储路径，找到并双击“R1.qpr”文件，即可执行并在【查询】窗口中看到运行结果。

**练习 2** 利用向导创建一个查询，其内容为“学生表.dbf”中姓名为“李丹洋”且入学成绩大于等于“600”的学生信息，并将该查询结果保存为一个名为“R2.qpr”的查询文件。

- (1) 执行【文件】→【新建】命令，并在系统弹出的【新建】对话框的【文件类型】区域中选择【查询】单选项，然后单击【向导】按钮。
- (2) 在弹出的【向导选取】对话框中，选择【查询向导】，然后单击【确定】按钮，弹出【查询向导】对话框。
- (3) 在向导的第 1 步中，单击【数据库和表】区域中的浏览按钮，并在打开的对话框中选择“学生表.dbf”所在的存储路径，找到并将其打开。

- (4) 仍在向导的第1步中, 双击【可用字段】列表框中右侧的【>>】按钮, 使全部字段均显示在【选定字段】列表框中。
- (5) 单击【下一步】按钮, 进入查询向导的第3步。参照图 4.30, 设置查询的筛选条件。(注意: 其值的内容是一个字符串, 在输入时, 需要在其文字两端加上西文半角状态下的双引号定界符"")。
- (6) 单击【完成】按钮, 弹出查询向导的第5步。再次单击【完成】按钮, 弹出【另存为】对话框。在其中选择适当的存储路径, 并在【文件名】文本框中输入“R2”, 然后单击【保存】按钮。

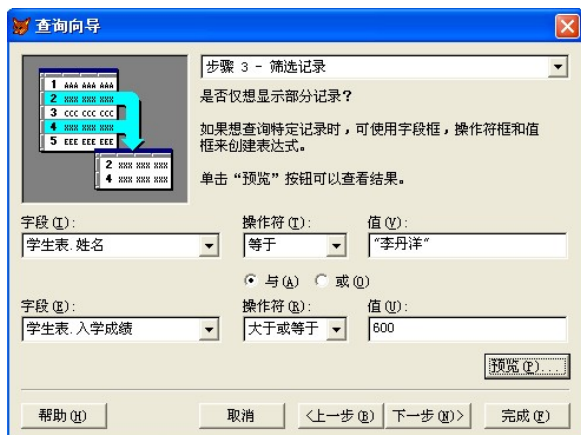


图 4.30 练习 2 的记录筛选设置

**练习 3** 利用向导创建一个查询, 其内容包含“学生表.dbf”中所有年龄大于等于 20 的学生信息, 查询结果按照年龄的升序显示, 并将该查询结果保存为一个名为“R3.qpr”的查询文件。将 R3.qpr 的查询结果输出到数据表中, 并命名为“A1.dbf”。

- (1) 执行【文件】→【新建】命令, 并在系统弹出的【新建】对话框的【文件类型】区域中选择【查询】单选项, 然后单击【向导】按钮。
- (2) 在弹出的【向导选取】对话框中, 选择【查询向导】, 然后单击【确定】按钮, 弹出【查询向导】对话框。
- (3) 在向导的第1步中, 单击【数据库和表】区域中的浏览按钮, 并在打开的对话框中选择“学生表.dbf”所在的存储路径, 找到并将其打开。
- (4) 仍在向导的第1步中, 双击【可用字段】列表框右侧的【>>】按钮, 使全部字段均显示在【选定字段】列表框中。
- (5) 单击【下一步】按钮, 进入查询向导的第3步, 设置查询的筛选条件。其中, 【字段】组合框中选择“出生日期”, 【操作符】组合框中为“小于或等于”, 【值】文本框中为 CTOD (“01/01/89”)。
- (6) 单击【下一步】按钮, 进入查询向导的第4步, 设置排序依据。双击【可用字段】列表框中的“学生表.出生日期”, 使其显示在右侧的【选定字段】列表框中。
- (7) 单击【完成】按钮, 弹出【另存为】对话框。在其中选择适当的存储路径, 并在【文件名】文本框中输入“R3”, 然后单击【保存】按钮。
- (8) 执行【文件】→【打开】命令, 弹出【打开】对话框。
- (9) 在对话框的【搜寻】列表框中选择“R3.qpr”所在的存储路径, 在【文件类型】中选择【查询】, 然后双击“R3.qpr”文件, 打开【查询设计器】窗口。

(10) 执行【查询】→【查询去向】命令，弹出【查询去向】对话框。

(11) 单击该对话框中的【表】按钮，并在弹出的【表名】文本框中输入“A1”，然后单击【确定】按钮。

**练习4** 利用查询设计器创建一个分组汇总查询“A2.qpr”，按性别分别统计出“学生表.dbf”中男、女学生的平均入学成绩。

(1) 执行【文件】→【新建】命令，并在系统弹出的【新建】对话框的【文件类型】区域中选择【查询】单选项，然后单击【新建文件】按钮。

(2) 在弹出的【打开】对话框中，选择要使用的数据表文件“学生表.dbf”，然后单击【确定】按钮。

(3) 在弹出的【添加表或视图】对话框中，单击【关闭】按钮，进入【查询设计器】窗口。

(4) 在【字段】选项卡中，双击【可用字段】列表框中的“学生表.性别”，使其显示在右侧的【选定字段】列表框中。

(5) 在【函数和表达式】文本框中直接输入或者单击其后面的浏览按钮，并在弹出的【表达式生成器】对话框中设置平均入学成绩的表达式：AVG(学生表.入学成绩)。

(6) 单击【添加】按钮，使该表达式也显示在右侧的【选定字段】列表框中，结果如图4.31所示。

(7) 选择【分组依据】选项卡，双击【可用字段】列表框中的“学生表.性别”，使其显示在右侧的【分组字段】列表框中，如图4.32所示。

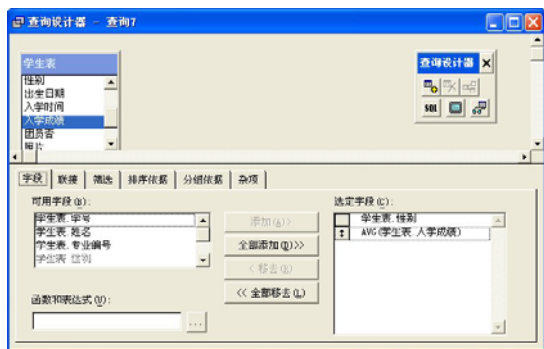


图 4.31 查询设计器的【字段】选项卡

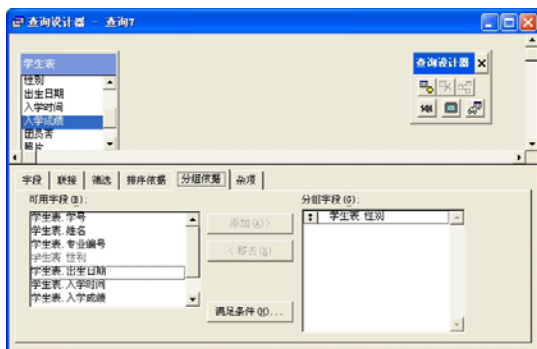


图 4.32 查询设计器【分组依据】选项卡

(8) 单击常用工具栏中的运行按钮，即可在弹出的【查询】窗口中看到执行结果，如图4.33所示。

(9) 关闭【查询设计器】窗口，并在弹出的【另存为】对话框中，选择适当的存储路径，在【文件名】文本框中输入“R4”，然后单击【保存】按钮。

性别	Avg_入学成绩
男	631.67
女	585.67

图 4.33 分组汇总的查询结果

**练习5** 利用查询设计器创建一个查询，要求显示出“学生表.dbf”中每个专业的最高入学成绩和最低入学成绩。

(1) 执行【文件】→【新建】命令，并在系统弹出的【新建】对话框的【文件类型】区域中选择【查询】单选项，然后单击【新建文件】按钮。

(2) 在弹出的【打开】对话框中，选择要使用的数据表文件“学生表.dbf”，然后单击【确定】按钮。

(3) 在弹出的【添加表或视图】对话框中，单击【关闭】按钮，进入【查询设计器】窗口。

(4) 在【字段】选项卡中，双击【可用字段】列表框中的“学生表.专业编号”，使其显示在右侧的【选定字段】列表框中。

- (5) 在【函数和表达式】文本框中直接输入或者单击其后面的浏览按钮，并在弹出的【表达式生成器】对话框中设置最高入学成绩的表达式：MAX(学生表.入学成绩)。
- (6) 单击【添加】按钮，使该表达式也显示在右侧的【选定字段】列表框中。
- (7) 重复步骤(5)和(6)，将最低入学成绩的表达式“MIN(学生表.入学成绩)”也添加到右侧的【选定字段】列表框中。
- (8) 选择【分组依据】选项卡，双击【可用字段】列表框中的“学生表.专业编号”，使其显示在右侧的【分组字段】列表框中。
- (9) 单击常用工具栏中的运行按钮，即可在弹出的【查询】窗口中看到执行结果。

**练习 6** 利用查询设计器创建一个查询，其内容包含数据库中所有女学生的学号、姓名、专业编号、课程号及其成绩的信息，要求查询结果按成绩的降序显示，并将查询结果命名为“学生成绩.qpr”。

- (1) 执行【文件】→【打开】命令，弹出系统的【打开】对话框。
- (2) 在对话框的【搜寻】选项中选择适当的存储路径，在【文件类型】区域中选择【数据库】然后双击“数据库.dbc”文件，打开【数据库设计器】窗口。
- (3) 执行【文件】→【新建】命令，并在系统弹出的【新建】对话框的【文件类型】区域中选择【查询】单选项，然后单击【新建文件】按钮。
- (4) 在弹出的【添加表或视图】对话框中，依次双击“学生表”和“学生选课表”两项，使它们全部显示在【查询设计器】窗口中。
- (5) 在弹出的【连接条件】对话框中，确认“学生表.学号—学生选课表.学号”的连接条件，以及“连接类型”为“内部连接”，然后单击【确定】按钮。
- (6) 在【添加表或视图】对话框中，单击【关闭】按钮，进入【查询设计器】窗口。
- (7) 在【字段】选项卡中，依次双击【可用字段】列表框中的“学号”、“姓名”、“专业编号”、“课程编号”、“成绩”，使它们显示在右侧的【选定字段】列表框中。
- (8) 在【筛选】选项卡中，设置查询的筛选条件。其中，【字段名】列表框中为“学生表.性别”，【条件】列表框中为“=”，【实例】列表框中为“女”。
- (9) 在【排序依据】选项卡中，先单击【排序选项】中的【降序】单选项，再双击【可用字段】列表框中的“学生选课表.成绩”，使其显示在右侧的【选定字段】列表框中。
- (10) 单击常用工具栏中的运行按钮，即可在弹出的【查询】窗口中看到执行结果。
- (11) 单击【查询设计器】窗口右上角的关闭按钮，并在弹出的“要将所做更改保存吗？”对话框中单击【是】按钮。
- (12) 在弹出的【另存为】对话框中，输入查询文件的名称“学生成绩”，并选择适当的存储路径，然后单击【保存】按钮。

## 4.2 视图

视图设计是为了便于用户对数据库进行操作，将同一数据库各表中的数据按要求重新组织成类同表供用户进一步操作，在 Visual FoxPro 中视图文件不能独立存在，它是数据库的一部分。根据视图的来源可分为本地视图和远程视图。本地视图的数据来自工作站，远程视图的数据来自数据服务器。



## 4.2.1 利用向导创建本地视图

下面通过一个实例利用向导创建本地视图。在“教师表”中，筛选出职称为讲师的男教师记录，要求只显示“教师编号”、“教师姓名”、“性别”、“职称”4个字段的内容。

### 1. 启动本地视图向导

在【项目管理器】窗口中选择【数据】选项卡中的【数据库】，在其下的选项中选择【本地视图】，单击【新建】按钮，屏幕显示【新建本地视图】对话框。

### 2. 选取字段

单击【新建本地视图】对话框中的【视图向导】按钮，屏幕显示【本地视图向导】对话框，并选取“教师表”中的“教师编号”、“教师姓名”、“性别”、“职称”4个字段，导入【选定字段】列表框中，如图 4.34 所示。

### 3. 筛选记录

单击【下一步】按钮，出现筛选记录的对话框，输入查询特定条件。在本例中选择性别等于“男”且职称等于“讲师”，如图 4.35 所示。



图 4.34 步骤 1-字段选取



图 4.35 步骤 3-筛选记录

### 4. 排序记录

单击【下一步】按钮，显示排序记录的对话框。在本例中选择“教师编号”为升序排序字段，添加到【选定字段】列表框中，如图 4.36 所示。

“步骤 4a-限制记录”和“步骤 5-完成”（如图 4.37 所示）的设置和使用与查询向导创建查询的步骤类似，在此不再重复叙述。

### 5. 保存本地视图

单击【完成】按钮，出现【视图名】对话框，要求给创建的视图命名一个文件名，本例中命名为“教师情况”。然后单击【确定】按钮，完成创建视图的操作，如图 4.38 所示。

在【项目管理器】窗口中，展开【本地视图】，可以观察到新创建的“教师情况”视图文件。选定新建的视图文件“教师情况”，单击【浏览】按钮，可看到视图运行结果，如图 4.39 所示。

可见，使用向导创建视图的操作过程与利用向导创建查询的操作过程是一样的。

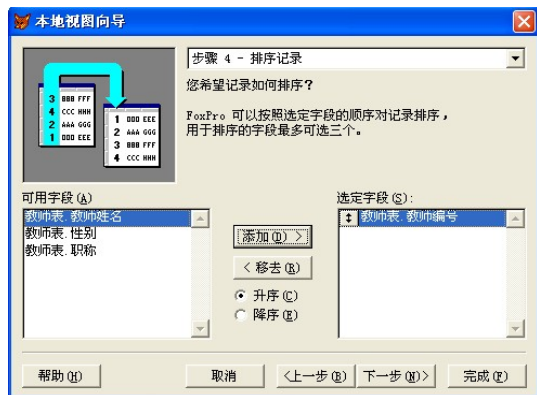


图 4.36 步骤 4-排序记录



图 4.37 步骤 5-完成

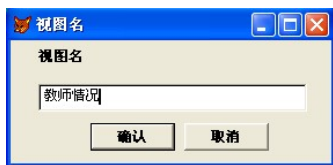


图 4.38 【视图名】对话框



图 4.39 视图预览结果

## 4.2.2 利用视图设计器创建本地视图

由于使用向导创建视图不能灵活自如地对记录进行分组、创建计算字段、更新数据等，因此常使用视图设计器来建立视图。这样用户可以自己设置更新条件、定义更新字段等。

下面通过一个实例利用视图设计器创建本地视图。在“教师表”中检索担任“VFP 数据库程序设计”、“大学计算机应用基础”且职称为“教授”、“副教授”的教师情况，要求只显示“教师表.教师编号”、“教师表.教师姓名”、“教师表.职称”、“教师任课表.课程编号”、“课程表.课程名称”、“课程表.学时”等字段内容。

### 1. 启动视图设计器

在【项目管理器】窗口中，选择【数据库】中的【本地视图】，单击【新建】按钮，显示【新建本地视图】对话框，然后单击【新建视图】按钮，屏幕显示【视图设计器】窗口。

### 2. 添加表或视图

在【视图设计器】窗口中，单击工具栏上的添加表按钮，屏幕显示【添加表或视图】对话框。在【添加表或视图】对话框中，选择要添加的表或视图。这里选择数据库中的“教师表”、“教师任课表”和“课程表”，如图 4.40 所示。



### 3. 建立表或视图之间的连接条件

当把“教师表”和“教师任课表”添加到【视图设计器】窗口后，如果这两个表之间没有显示一条关系连线，这时应在这两个表中分别选择一个字段，如利用“教师编号”来建立两表之间的连接。建立两表之间的连接方法可以通过以下几种方式来完成。

① 按住鼠标左键不放，将“教师表.教师编号”字段拖到“教师任课表.教师编号”字段上，两个字段之间显示一条连线，表示两表之间的连接关系已经建立了。

② 单击【连接】选项卡，输入连接条件，如本例中的“教师表.教师编号 = 教师任课表.教师编号”。单击【视图设计器】窗口工具栏上的添加连接按钮，利用【连接条件】对话框进行两表之间的连接，如图 4.41 所示。

③ 系统自动弹出【连接方式】对话框。

### 4. 选择显示字段

根据实例的要求，单击【视图设计器】窗口中的【字段】选项卡，分别将“教师表.教师编号”、“教师表.教师姓名”、“教师表.职称”、“教师任课表.课程编号”、“课程表.课程名称”、“课程表.学时”等字段添加到【选定字段】列表框中。



图 4.40 【视图设计器】窗口中添加的表

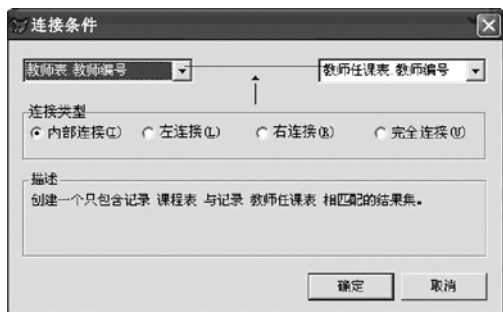


图 4.41 【连接条件】对话框

### 5. 设定筛选条件

在运行视图时，符合筛选条件的记录才显示到屏幕上。这就需要通过筛选条件来确定检索哪些符合条件的记录，本例中检索担任“VFP 数据库程序设计”、“大学计算机应用基础”且职称为“教授”、“副教授”的教师情况，因此在【视图设计器】窗口的【筛选】选项卡中输入：

```
教师表.职称 IN "教授","副教授" AND  
课程表.课程名称 IN "VFP 数据库程序设计", "大学计算机应用基础"
```

输入上述条件后就确定了筛选条件，如图 4.42 所示。单击运行按钮浏览结果，如图 4.43 所示。

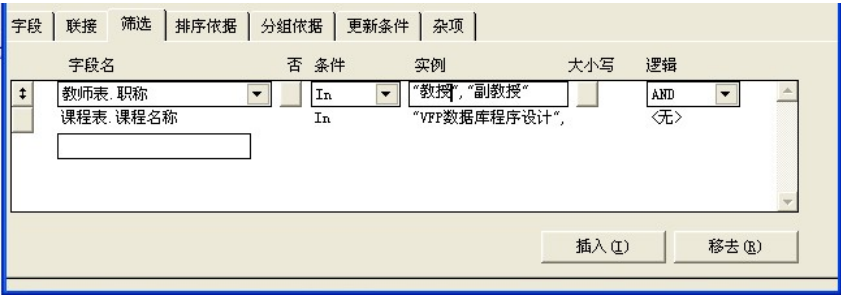


图 4.42 设定筛选条件

教师编号	教师姓名	职称	课程编号	课程名称	学时
CC0101	赵晓兰	教授	CS02	大学计算机应用基础	30
CC0404	刘雪燕	副教授	CS01	VFP数据库程序设计	75
CS0101	林星明	教授	CS02	大学计算机应用基础	30
CS0202	赵华林	副教授	CS02	大学计算机应用基础	30
JS0101	秦健中	副教授	CS01	VFP数据库程序设计	75
JS0202	吴蕴敏	副教授	CS01	VFP数据库程序设计	75

图 4.43 浏览结果

6. 设定更新条件

视图的最大特点是可以对源表中的数据进行更新。要更新数据就必须设置更新条件。这时需要定义一个关键字字段，系统根据关键字字段列出原数据文件中与之对应的记录，并进行修改操作。关键字字段的设置必须是唯一的，若有重复值，则必须选取多个关键字字段来避免重复，否则系统将无法判断要修改的记录。

下面我们来看如何在已创建的视图中，更新“教师表”中的“教师姓名”、“职称”和“课程表”中的“学时”字段。

在【视图设计器】窗口中，利用【更新条件】选项卡来设置视图中的关键字字段。设置的方法：在【更新条件】选项卡中，用鼠标单击“字段名”左边的复选按钮，即带有钥匙图标的一列，然后会在相应的字段前画一个“√”。根据要求设置“教师表”中的“教师编号”和“课程表”中的“课程名称”为关键字字段。

接下来设置可更新字段。其方法是在【更新条件】选项卡中，单击“字段名”左边的复选按钮，即带有铅笔图标的一列，然后会在相应的字段前画一个“√”。根据本例的要求，将“教师表”中的“职称”、“课程表”中的“学时”两个字段设置为可更新的字段，如图 4.44 所示。

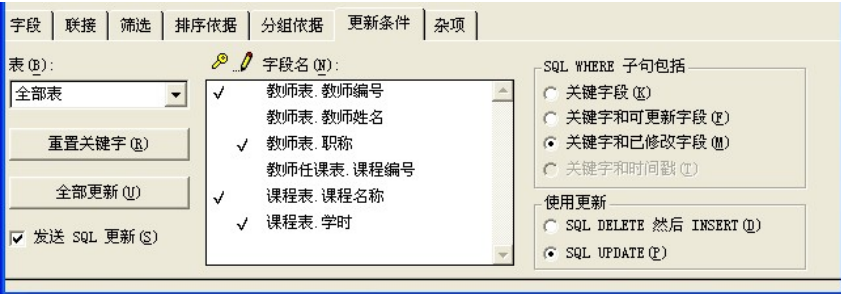


图 4.44 【更新条件】选项卡

提示:

- ① 由于关键字字段是用于唯一标识每一条记录的, 所以最好不要使用关键字字段作为更新的字段。
- ② 如果要将表中修改的记录回存到原表中, 必须选中【发送 SQL 更新】选项。在使用此项之前, 必须至少设置一个关键字字段和一个可修改的字段。

## 7. 更新记录

“更新条件”设置完后, 即可在浏览窗口中修改可更新的字段, 并将更新后的结果回存到原表中。现将“教师表”中的“姓名”、“职称”两个字段设置为可更新的字段。运行该视图后, 在浏览窗口中将记录的教师姓名为“吴莲敏”的职称改为“副教授”, 并将结果回存到表中, 如图 4.45 所示。

教师编号	教师姓名	性别	职称	毕业学校	课程编号	工资	政府津贴
JC0101	陈得江	男	教授	北京大学	CC01	3310.10	T
JC0202	赵惠兰	女	副教授	清华大学	CC01	2587.00	F
JC0303	刘大林	男	讲师	上海交通大学	CC02	2000.00	F
JS0101	秦健中	男	副教授	浙江大学	CS01	3120.00	T
JS0202	吴莲敏	女	副教授	复旦大学	CS01	1500.00	F
CC0101	赵晓兰	女	一级教授	同济大学	CS02	3334.00	F
CC0202	李月敏	女	讲师	吉林大学	CS02	1678.00	F
CC0303	王克强	男	讲师	北京理工大学	CC03	1564.00	F
CC0404	刘雪燕	女	副教授	北京交通大学	CS01	2345.00	F
CS0101	林星明	男	教授	北京林业大学	CS02	3254.00	F
CS0202	赵华林	男	副教授	厦门大学	CS02	2766.00	F
CS0303	李严冬	男	教授	哈尔滨工业大学	CC03	3786.00	T

图 4.45 修改后的原表结果

## 8. 保存视图

单击工具栏中的保存按钮, 在打开的【保存】对话框中, 输入视图文件名: “教师任课视图”, 单击【确定】按钮。

## 9. 运行视图

单击【查询】菜单中的【运行查询】命令或单击工具栏中的运行按钮, 即可运行该视图文件。

## 10. 查看SQL语句

单击【查询】菜单中的【查看 SQL】命令, 可以看到用 SQL 语言生成的语句代码, 如图 4.46 所示。

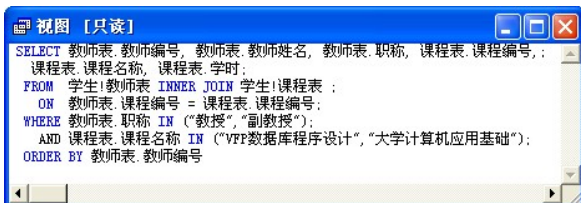


图 4.46 SQL 语言生成的语句代码

### 4.2.3 创建参数化视图

前面所设置的筛选条件都是固定的，如上例中建立的条件表达式：教师表.职称 = "教授"。每次运行视图时，所显示的记录都是职称为教授的记录，这时如要检索其他职称的教师，则需要修改视图中的筛选条件，非常不方便。系统提供了设置视图参数的功能来克服这个缺点，每次运行视图时根据输入值的不同而产生不同的查询结果，如根据编号查询任意职称的教师情况。

选择“教师情况”视图，单击【修改】按钮，在【视图设计器】窗口中打开前面新创建的视图。在【视图设计器】窗口中，单击【查询】菜单中的【视图参数】命令，屏幕的出现【视图参数】对话框。在该对话框的【参数名】中输入一个参数名称，例如，输入“职称”，在【类型】下拉列表框中选择参数的类型，这里选择为“字符型”，如图 4.47 所示。

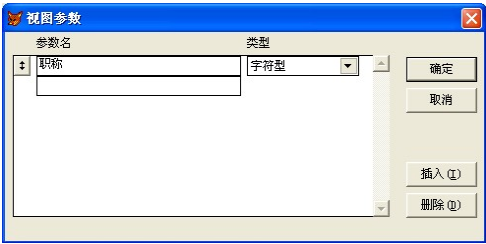


图 4.47 【视图参数】对话框

接着在【视图设计器】窗口中，单击【筛选】选项卡，选择适当的筛选字段和筛选条件后，在【实例】栏中，输入一个“?”紧接一个参数名。例如，本例中的筛选字段为“教师表.职称”，筛选条件为“=”，实例为“?职称”，如图 4.48 所示。



图 4.48 设置筛选条件的对话框

单击工具栏中的运行按钮，系统会弹出【视图参数】对话框，如图 4.49 所示。输入参数“副教授”，单击【确定】按钮。此时系统会显示职称为“副教授”的教师情况，如图 4.50 所示。

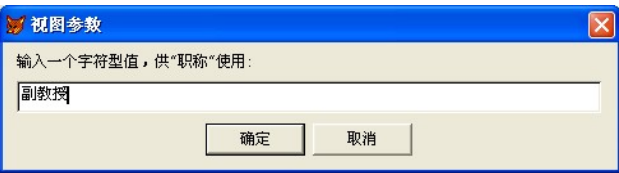


图 4.49 【视图参数】对话框

教师情况			
教师编号	教师姓名	性别	职称
CS0202	赵华林	男	副教授
JS0101	秦健中	男	副教授

图 4.50 参数视图查询结果

命名该视图后进行保存，这样就完成了一个参数化视图。用户可以设置多个参数，根据不同的值检索出一组不同的记录，而不需要重新建立视图。

## 4.2.4 创建远程视图

本地视图使用 Visual FoxPro 中的数据库或数据表中的数据，而远程视图则使用远程 ODBC 数据源中选取的数据。在建立远程视图之前，必须先连接到一个远程数据源上。从 ODBC 服务器上提取一部分数据，在本地对所选择的记录进行更改或者添加后，其结果可以返回到远程的数据源上。

利用视图设计器或远程视图向导建立远程视图的方法同建立本地视图的方法基本相同，只是在数据源的选取上有所不同。

### 1. 与远程数据连接

从 ODBC 服务器上提取一部分数据，在本地对所选择的记录进行更改或添加后，将其结果返回到远程的数据源上。建立远程数据连接的操作步骤如下。

- (1) 从【项目管理器】窗口中选择【数据库】，然后单击【连接】选项卡。
- (2) 单击【新建】按钮，出现【连接设计器】窗口。
- (3) 在【连接设计器】窗口中，按要求在【数据源】列表框中选择数据源。
- (4) 如果有必要输入“用户标识”、“密码”和“数据库”项。
- (5) 指定显示 ODBC 登录提示的时刻，设置数据处理方式，设置超时间隔。单击【验证连接】按钮，可对刚输入内容的连接进行检查。如果连接成功，则显示对话框提示成功信息；如果连接失败，则出现错误信息。
- (6) 利用【文件】菜单的【保存】命令，命名此连接并保存。

在【连接设计器】窗口的【数据源】列表框中，如果没有需要的数据源，可以单击【新的数据】按钮后进入 ODBC 数据源管理器来设置数据源，也可以单击【验证连接】按钮来验证连接是否能执行。也可以通过【文件】菜单中的【新建】命令来建立一个新的连接。

### 2. 创建远程视图

在建立有效的数据源或与远程数据连接后，就可使用项目管理器来创建远程视图了。远程视图与本地视图类似，只是在定义它时需要加入连接名称或数据源名称。下面我们介绍利用远程视图设计器创建远程视图的方法。

- (1) 在【项目管理器】窗口中的一个数据库下选择【远程视图】，然后单击【新建】按钮，打开【选择连接或数据源】对话框，如图 4.51 所示。
- (2) 在【选择连接或数据源】对话框中，选择可用的数据源或命名连接并且单击【确定】按钮。
- (3) 如果需要，单击【新建】按钮，系统会弹出【连接设计器】对话框，提示输入用户标识和口令，如图 4.52 所示。

连接成功后，系统将弹出【选择远程数据库中的表】对话框。选择表后，将显示视图设计器。使用视图设计器创建远程视图与创建本地视图的其他步骤相同。

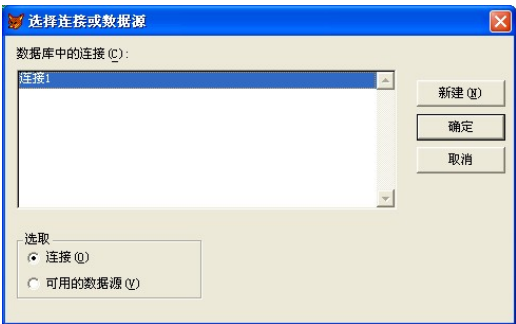


图 4.51 【选择连接或数据源】对话框

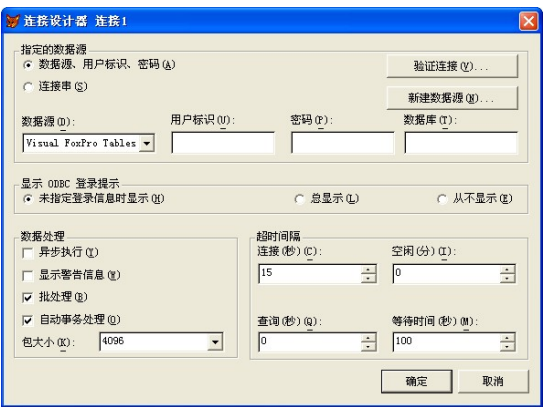


图 4.52 【连接设计器】对话框

数据库的应用大大提高了对数据的处理能力，这一章主要论述了怎样对数据库中的数据进行查询和更新。怎样使用视图来协助从本地或远程数据源中获取相关数据，可以对这些数据进行修改并更新，Visual FoxPro 将自动完成对源表的更新。还可以实现对多个表生成一个“查询”或“视图”，这些表既可以是本地的，也可以是远程的。

练习与操作

练习 1 利用向导创建一个本地视图，其内容包含：“学生表”中所有学生的学号、姓名、专业编号和入学成绩，并将该视图命名为“ST1”。查看视图“ST1”中内容。在视图“ST1”中，取消“专业编号”字段的信息。参考步骤如下。

- (1) 执行【文件】→【打开】命令，弹出【打开】对话框。
- (2) 在对话框的【搜寻】列表框中选择“数据库.dbc”所在的存储路径，在【文件类型】区域中选择【数据库】单选项，然后双击“数据库.dbc”文件，打开【数据库设计器】窗口。
- (3) 执行【文件】→【新建】命令，并在弹出的【新建】对话框的【文件类型】区域中选择【视图】单选项，然后单击【视图向导】按钮，弹出【本地视图向导】对话框。
- (4) 在向导的步骤 1 中，单击选中【数据库和表】列表框中的“学生表”，然后在【可用字段】列表框中依次双击“学号”、“姓名”、“专业编码”、“入学成绩”4 项，使其显示在右侧的【选定字段】列表框中。
- (5) 单击【完成】按钮，弹出视图向导的步骤 5。
- (6) 再次单击【完成】按钮，弹出【视图名】对话框。在其中输入“ST1”，然后单击【确认】按钮，即可返回【数据库设计器】窗口，从中可以看到新添加的视图“ST1”，如图 4.53 所示。

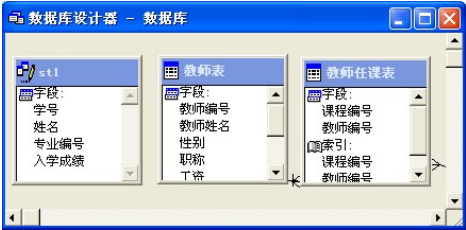


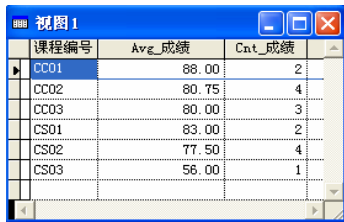
图 4.53 【数据库设计器】窗口



- (7) 在【数据库设计器】窗口中，用鼠标双击视图“ST1”小窗口的标题栏，即可查看该视图的内容。
- (8) 在【数据库设计器】窗口中，用鼠标右键单击视图“ST1”小窗口的标题栏，选择其快捷菜单中的【修改】命令，打开【视图设计器】窗口。
- (9) 在【字段】选项卡中，双击【选定字段】列表框中的“学生表.专业编号”，使其回到左侧的【可用字段】列表框中。
- (10) 单击【查询设计器】窗口右上角的关闭按钮，并在弹出的询问“要将所做修改保存到 ST1 中吗？”的对话框中单击【是】按钮，即可将所做修改保存后返回【数据库设计器】窗口。

**练习 2** 利用视图设计器创建一个本地视图，其内容包含“学生选课表”中每门课程的平均分数，并将该视图命名为“ST2”。参考步骤如下。

- (1) 在【数据库设计器】窗口中，打开“学生选课库”数据库。
- (2) 执行【文件】→【新建】命令，并在弹出的【新建】对话框的【文件类型】区域中选择【视图】单选项，然后单击【新建文件】按钮。
- (3) 在弹出的【添加表或视图】对话框中，双击“学生选课表”，使它显示在【视图设计器】窗口中。
- (4) 在【添加表或视图】对话框中，单击关闭按钮，进入【视图设计器】窗口。
- (5) 在【字段】选项卡中，双击【可用字段】列表框中的“学生选课表.课程编号”，使其显示在右侧的【选定字段】列表框中。
- (6) 在【函数和表达式】文本框中直接输入或者单击其后面的按钮，并在弹出的【表达式生成器】对话框中设置平均分数的表达式：AVG(学生选课表.成绩)。
- (7) 单击【添加】按钮，使该表达式也显示在右侧的【选定字段】列表框中。
- (8) 重复步骤(6)和(7)，将统计考试人次的表达式“COUNT(学生选课表.成绩)”也添加到右侧的【选定字段】列表框中。
- (9) 选择【分组依据】选项卡，双击【可用字段】列表框中的“学生选课表.课程编号”，使其显示在右侧的【分组字段】列表框中。
- (10) 单击常用工具栏中的运行按钮，即可在弹出的【视图 1】窗口中看到执行结果，如图 4.54 所示。
- (11) 单击【查询设计器】窗口右上角的关闭按钮，在弹出的询问“要将所做修改保存吗？”的对话框中单击【是】按钮。然后，在弹出的【保存】对话框中输入视图名称“ST2”，单击【确认】按钮，即可返回【数据库设计器】窗口，并在其中看到新添加的视图“ST2”。



课程编号	Avg_成绩	Cnt_成绩
CC01	88.00	2
CC02	80.75	4
CC03	80.00	3
CS01	83.00	2
CS02	77.50	4
CS03	56.00	1

图 4.54 视图运行结果

**练习 3** 利用设计器创建一个本地视图，其内容包含“学生选课库”数据库所有选修课程“CC02”的学生的学号、姓名及其成绩的信息，要求其结果按成绩的降序显示，并且允许修改成绩，并将该视图命名为“ST3”。在视图“ST3”中修改某学生的成绩，然后验证其更新源表的功能。

- (1) 在【数据库设计器】窗口中，打开“学生选课库”数据库。
- (2) 执行【文件】→【新建】命令，并在系统弹出的【新建】对话框的【文件类型】区域中选择【视图】单选项，然后单击【新建文件】按钮。
- (3) 在弹出的【添加表或视图】对话框中，依次双击“学生表”和“学生选课表”两项，使它们全部显示在【视图设计器】窗口中。

- (4) 在弹出的【连接条件】对话框中，确认“学生表.学号—学生选课表.学号”的连接条件，并将“连接类型”设为“内部连接”，然后单击【确定】按钮。
- (5) 在【添加表或视图】对话框中，单击关闭按钮，进入【视图设计器】窗口。
- (6) 在【字段】选项卡中，依次双击【可用字段】列表框中的“学号”、“姓名”、“成绩”3项，使它们显示在右侧的【选定字段】列表框中。
- (7) 在【筛选】选项卡中，设置查询的筛选条件。其中，“字段名”为“学生选课表.课程编号”，“条件”为“=”，“实例”为“CC02”。
- (8) 在【排序依据】选项卡中，先单击【排序选项】中的【降序】单选项，再双击【可用字段】列表框中的“学生选课表.成绩”，使其显示在右侧的【选定字段】列表框中。
- (9) 选择【视图设计器】窗口中的【更新条件】选项卡，在其“字段名”左侧中，选中“学生学号”前面的钥匙图标，将“学生学号”字段设置为关键字字段。
- (10) 在“字段名”左侧中，先选中“学生选课表.成绩”前面的钥匙图标，再单击后面的铅笔图标，将“成绩”字段设置为允许修改状态。
- (11) 单击选中窗口左下角的【发送 SQL 更新】复选框。
- (12) 单击常用工具栏中的运行按钮，即可在弹出的【查询】窗口中看到执行结果。
- (13) 单击【查询设计器】窗口右上角的关闭按钮，在弹出的询问“要将所做修改保存吗?”的对话框中单击【是】按钮。然后，在弹出的【保存】对话框中输入视图名称“ST3”，单击【确认】按钮，即可返回【数据库设计器】窗口，可以从中看到新添加的视图“ST3”。
- (14) 在【数据库设计器】窗口中，使用鼠标双击视图“ST3”小窗口的标题栏，打开视图浏览窗口。
- (15) 任选一条记录，更改其成绩数据。
- (16) 关闭视图浏览窗口，再双击“学生选课库”小窗口的标题栏，即可在打开的表浏览窗口中看到相应记录的成绩已被自动更新。

## 4.3 关系数据库标准语言SQL

结构化查询语言 SQL (Structured Query Language) 是一种介于关系代数与关系演算之间的语言，其功能包括查询、操纵、定义和控制 4 个方面。SQL 是一个通用的、功能极强的关系数据库语言，目前已成为关系数据库的标准语言。

### 4.3.1 SQL概述

#### 1. SQL的主要特点

SQL语言之所以能够为用户和业界所接受，成为国际标准，是因为它是一个综合的、通用的、功能极强同时又简洁易学的语言。SQL语言集 数据查询 (data query)、数据操纵 (data manipulation)、数据定义 (data definition) 和 数据控制 (data control) 功能于一体，充分体现了关系数据库语言的特点和优点。SQL主要包括以下几个特点。

##### 1) 综合统一

由于SQL语言集数据定义语言 (DDL)、数据操纵语言 (DML)、数据控制语言 (DCL) 的功能于一体，语言风格统一，可以独立完成数据库生命周期中的全部活动，包括定义关系



模式、录入数据以建立数据库、查询、更新、维护、数据库重构、数据库安全性控制等一系列操作，这就为数据库应用系统开发提供了良好的环境。例如，用户在数据库投入运行后，还可根据需要随时地修改模式，并不影响数据库的运行，从而使系统具有良好的可扩充性。

## 2) 高度非过程化

非关系数据模型的数据操纵语言是面向过程的语言，用其完成某项请求，必须指定存取路径。而用SQL语言进行数据操作，用户只需提出“做什么”，而不必指明“怎么做”，因此用户无须了解存取路径。存取路径的选择及SQL语句的操作过程由系统自动完成。这不但大大减轻了用户负担，而且有利于提高数据独立性。

## 3) 面向集合的操作方式

SQL语言采用集合操作方式，不仅查找结果可以是元组的集合，而且一次插入、删除、更新操作的对象也可以是元组的集合。

非关系数据模型采用的是面向记录的操作方式，任何一个操作其对象都是一条记录。例如，查询所有平均成绩在60分以上的学生姓名，用户必须说明完成该请求的具体处理过程，即如何用循环结构按照某条路径一条条地把满足条件的学生记录读取出来。

## 4) 以同一种语法结构提供两种使用方式

SQL语言既是自含式语言，又是嵌入式语言。作为自含式语言，它能够独立地用于联机交互的使用方式，用户可以在终端键盘上直接输入SQL命令对数据库进行操作。作为嵌入式语言，SQL语句能够嵌入到高级语言（如C、COBOL、FORTRAN、PL/1）程序中，供程序员在设计程序时使用。而在两种不同的使用方式下，SQL语言的语法结构基本上是一致的。这种以统一的语法结构提供两种不同的使用方式的特点，为用户提供了极大的灵活性与方便性。

## 5) 语言简洁，易学易用

SQL语言功能极强，但由于设计巧妙，语言十分简洁，完成数据定义、数据操纵、数据控制的核心功能只用了9个动词：CREATE、DROP、ALTER、SELECT、INSERT、UPDATE、DELETE、GRANT、REVOKE，如表4.1所示。而且SQL语言语法简单，接近英语口语，因此易学易用。

表 4.1 SQL 的常用命令

SQL 功能	命令动词
数据查询	SELECT
数据定义	CREATE、DROP、ALTER
数据操纵	INSERT、UPDATE、DELETE
数据控制	GRANT、REVOKE

# 2. SQL语言的基本概念

## 1) SQL语言的模式结构

SQL语言支持关系数据库的三级模式结构。其中外模式对应于视图（view）和部分基本表（base table），模式对应于基本表，内模式对应于存储文件。

① 基本表是本身独立存在的表，在 SQL 中一个关系就对应一个表。一些基本表对应一个存储文件，一个表可以带有若干索引，索引也存放在存储文件中。

② 存储文件的逻辑结构组成了关系数据库的内模式。存储文件的物理文件结构是任意的。

③ 视图是从基本表或其他视图中导出的表，它本身不独立存储在数据库中，也就是说，数据库中只存放视图的定义而不存放视图对应的数据，这些数据仍存放在导出视图的基本表中，显然视图是一个虚表。用户可以用 SQL 语言对视图和基本表进行查询。从用户的角度来看，视图和基本表都是关系，而存储文件对用户是透明的。

## 2) SQL语言的使用

SQL 语言可以直接以命令方式交互使用，也可以嵌入到程序设计语言中以程序方式使用，VFP 采用了后者。VFP 在 SQL 方面提供了数据定义、数据查询和数据操纵功能，但没有提供数据控制功能。

## 3. 查询功能

SQL 语言的核心是查询。SQL 的查询命令也称为 SELECT 命令，它的基本形式由 SELECT-FROM-WHERE 查询模块组成，多个查询可以嵌套执行。VFP 的 SELECT 命令的语法规则如下：

```
SELECT  字段列表 FROM  表列表
      [WHERE  <条件表达式>]
      [GROUP BY...]
      [HAVING<条件表达式>]
      [UNION...]
      [ORDER BY...]
```

SELECT 查询命令的功能就是构造各种各样的查询，具体来说：

- SELECT 说明要查询的数据。
- FROM 说明要查询的数据来自哪个或哪些表，可以对单个表或多个表进行查询。
- WHERE 说明查询条件，即选择元组的条件。
- GROUP BY 短语用于对查询结果进行分组，可以利用它进行分组汇总。
- HAVING 短语必须跟随 GROUP BY 短语使用，它用来限定分组必须满足的条件。
- ORDER BY 用来对查询的结果进行排序。

## 4.3.2 SQL的各类查询

本章给出的 SQL 各类查询的例子均参照图 4.55~图 4.58 所示的关系表来完成。

### 1. 简单查询

简单查询是 SQL 语言中最简单的查询操作，这些查询都基于单个表，可以带有简单的条件。由 SELECT-FROM 短语构成无条件查询，或由 SELECT-FROM-WHERE 短语构成条件查询。下面是几个简单查询的例子。

供应商号	供应商名	地址
S3	新悦电子厂	西安
S4	华贸电子厂	北京
S6	新郑电子厂	郑州
S7	爱民电子厂	北京

图 4.55 “供应商”表

职工号	供应商号	订购单号	订购日期
A3	S7	OR67	06/28/04
A1	S4	OR73	07/28/04
A7	S4	OR76	05/25/04
A6		OR77	/ /
A3	S4	OR79	06/13/04
A1		OR80	/ /
A3		OR80	/ /
A3	S3	OR91	07/31/04

图 4.56 “订购单”表

仓库号	城市	面积
SH1	北京	1370
SH2	上海	1500
SH3	广州	1200
SH4	武汉	1400

图 4.57 “仓库”表

仓库号	职工号	工资
SH2	A1	2220
SH1	A3	2210
SH2	A4	2250
SH3	A6	2230
SH1	A7	2250

图 4.58 “职工”表

例 4.1 从职工关系中检索所有的工资值。

```
SELECT 工资 FROM 职工
```

结果是

```
2220
2210
2250
2230
2250
```

可见在运行结果中有重复值，如果要去掉查询结果中的重复值，需指定 DISTINCT 短语：

```
SELECT DISTINCT 工资 FROM 职工
```

例 4.2 检索仓库关系中的所有元组。

```
SELECT * FROM 仓库
```

结果是

```
SH1 北京 1370
SH2 上海 1500
SH3 广州 1200
SH4 武汉 1400
```

式中的 “\*” 是通配符，表示所有字段，所以上式等价于：

```
SELECT 仓库号, 城市, 面积 FROM 仓库
```

例 4.3 检索工资多于 2220 元的职工号。

```
SELECT 职工号 FROM 职工 WHERE 工资>2220
```

结果是

```
A4
A6
A7
```

**例 4.4** 检索哪些仓库有工资多于 2220 元的职工。

```
SELECT DISTINCT 仓库号 FROM 职工 WHERE 工资 > 2220
```

结果是

```
SH1  
SH2  
SH3
```

**例 4.5** 给出仓库在“SH1”或“SH2”工作，并且工资少于 2250 元的职工。

```
SELECT 职工号 FROM 职工;      && 这里的分号指的是续行符号  
WHERE 工资 < 2250 AND (仓库号 = "SH1" OR 仓库号 = "SH2")
```

结果是

```
A1  
A3
```

以上的检索只是基于一个关系。如果要想基于多个关系进行查询，则要进行连接查询。下面介绍连接查询。

## 2. 简单连接查询

前面介绍过，连接是关系的基本操作之一，连接查询是一种基于多个关系的查询。下面介绍几个简单连接查询的例子。

**例 4.6** 找出工资多于 2230 元的职工号和他们所在的城市。

```
SELECT 职工号, 城市 FROM 职工, 仓库;  
WHERE (工资 > 2230) AND (职工.仓库号 = 仓库.仓库号)
```

结果是

```
A4  上海  
A7  北京
```

这里的“职工.仓库号 = 仓库.仓库号”是连接条件。仓库关系和职工关系之间存在一个一对多的联系。

**例 4.7** 找出工作在面积大于 1400 的仓库的职工号及这些职工所在的城市。

```
SELECT 职工号, 城市 FROM 职工, 仓库;  
WHERE (面积 > 1400) AND (职工.仓库号 = 仓库.仓库号)
```

结果是

```
A1  上海  
A4  上海
```

## 3. 嵌套查询

基于多个关系的查询，这类查询所要求的结果出自一个关系，但相关的条件却涉及多个关系。如果在检索关系 X 中的元组时，它的条件依赖于相关的关系 Y 中的元组属性值，这时要使用嵌套查询。下面来看几个例子。

**例4.8** 查找哪些城市至少有一个仓库的职工工资为 2250 元。此例要求查询仓库表中的城市信息，而查询条件是职工表中的工资字段值。可得到以下嵌套查询：

```
SELECT 城市 FROM 仓库 WHERE 仓库号 IN;  
(SELECT 仓库号 FROM 职工 WHERE 工资 = 2250)
```

结果是

北京  
上海

这个命令中有两个 SELECT-FROM-WHERE 查询块，通常称为内层查询块和外层查询块，在该例中内层查询块检索到的仓库号值是 SH1 和 SH2，也可写出如下等价的命令：

```
SELECT 城市 FROM 仓库 WHERE 仓库号 IN("SH1","SH2")
```

这里的 IN 相当于集合运算符  $\in$ 。

**例4.9** 查询所有的职工工资都多于 2210 元的仓库的信息。此例也可描述为查询没有一个职工的工资少于或等于 2210 元的仓库的信息。可有以下的 SQL 命令：

```
SELECT * FROM 仓库 WHERE 仓库号 NOT IN;  
(SELECT 仓库号 FROM 职工 WHERE 工资 <= 2210)
```

结果是

SH2 上海 1500  
SH3 广州 1200  
SH4 武汉 1400

如果要排除那些还没有职工的仓库，检索要求描述为查询所有的职工工资都多于 2210 元的仓库的信息，并且该仓库至少要有一名职工。可写出 SQL 命令：

```
SELECT * FROM 仓库 WHERE 仓库号 NOT IN;  
(SELECT 仓库号 FROM 职工 WHERE 工资 <= 2210);  
AND 仓库号 IN (SELECT 仓库号 FROM 职工)
```

这样在内层查询中有两个并列的查询，得到的结果中将不包含没有职工的仓库信息。结果是

SH2 上海 1500  
SH3 广州 1200

**例 4.10** 找出与职工 A4 挣同样工资的所有职工。

```
SELECT 职工号 FROM 职工 WHERE 工资 =;  
(SELECT 工资 FROM 职工 WHERE 职工号 ="A4")
```

结果是

A4  
A7

## 4. 排序查询

使用 SQL 的 SELECT 语句可以对查询结果进行排序，具体格式为

```
ORDER BY, Order_Item [ASC|DESC][, Order_Item [ASC|DESC]...]
```

例 4.11 按职工的工资值升序检索出全部职工信息。

```
SELECT * FROM 职工 ORDER BY 工资
```

结果是

```
SH1  A3  2210
SH2  A1  2220
SH3  A6  2230
SH2  A4  2250
SH1  A7  2250
```

如果按降序排列，应加上 DESC:

```
SELECT * FROM 职工 ORDER BY 工资 DESC
```

例 4.12 先按仓库号排序, 再按工资排序并输出全部职工信息。

```
SELECT * FROM 职工 ORDER BY 仓库号, 工资
```

结果是

```
SH1  A3  2210
SH1  A7  2250
SH2  A1  2220
SH2  A4  2250
SH3  A6  2230
```

提示: ORDER BY 是对最终的查询结果进行排序, 不可以在子查询中使用该短语。

## 5. 简单的计算查询

SQL 不仅具有查询能力, 而且还有计算方式的检索, 用于计算检索的函数有

- COUNT——计数
- SUM——求和
- AVG——计算平均值
- MAX——求最大值
- MIN——求最小值

这些函数可以用在 SELECT 短语中对查询结果进行计算。

例 4.13 找出供应商所在地的数目。

```
SELECT COUNT (DISTINCT 地址) FROM 供应商
```

因为可以查询出北京、西安和郑州三个地址, 所以结果为 3。

提示: 除非对关系中的元组个数进行计数, 一般 COUNT 函数应该使用 DISTINCT。例如:

```
SELECT COUNT(*) FROM 供应商
```

这个命令将给出供应商关系中的记录数为 4。

**例 4.14** 求支付的工资总数。

```
SELECT SUM(工资) FROM 职工
```

 && 该命令将求出职工关系中工资的总和。

结果是

```
11160
```

**例 4.15** 求北京和上海的仓库职工的工资总和。

```
SELECT SUM(工资) FROM 职工 WHERE 仓库号 IN;  
(SELECT 仓库号 FROM 仓库 WHERE 城市="北京" OR 城市="上海")
```

结果是

```
8930
```

**例 4.16** 求所有职工的工资都多于 2210 元的仓库的平均面积。

```
SELECT AVG(面积) FROM 仓库 WHERE 仓库号 NOT IN ;  
(SELECT 仓库号 FROM 职工 WHERE 工资<=2210)
```

结果是

```
1366.67
```

**提示：**以上结果包含了尚没有职工的 SH4 仓库。

如果要排除没有职工的仓库，以上语句应改为

```
SELECT AVG(面积) FROM 仓库 WHERE 仓库号 NOT IN ;  
(SELECT 仓库号 FROM 职工 WHERE 工资<=2210);  
AND 仓库号 IN (SELECT 仓库号 FROM 职工)
```

结果是

```
1350
```

**例 4.17** 求在 SH2 仓库工作的职工的最高工资值。

```
SELECT MAX(工资) FROM 职工 WHERE 仓库号 = "SH2"
```

结果是

```
2250
```

如果要求该条件下的最低工资值，命令如下：

```
SELECT MIN(工资) FROM 职工 WHERE 仓库号="SH2"
```

结果是

```
2220
```

## 6. 分组与计算查询

在 SQL 中可以利用 GROUP BY 进行分组计算查询，GROUP BY 命令的格式如下：

```
GROUP BY GroupColumn [,GroupColumn...][HAVING FilterCondition]
```

其功能是将查询结果按一列或多列分组，如果未对查询结果分组，命令将作用于整个查询结果。分组后集函数将作用于每一个组，即每一组都有一个函数值。其中，HAVING 子句用于进一步限定分组条件，HAVING 子句总是跟在 GROUP BY 子句之后，不可以单独使用。HAVING 子句和 WHERE 子句并不矛盾，在查询中是先用 WHERE 子句限定元组，然后进行分组，最后再用 HAVING 子句限定分组。

**例 4.18** 求每个仓库的职工的平均工资。

```
SELECT 仓库号, AVG(工资) FROM 职工 GROUP BY 仓库号
```

结果是

```
SH1  2230.00
SH2  2235.00
SH3  2230.00
```

**例 4.19** 求至少有两个职工的每个仓库的平均工资。

```
SELECT 仓库号, COUNT(*), AVG(工资) FROM 职工;
GROUP BY 仓库号 HAVING COUNT(*) >= 2
```

结果是

```
SH1  2  2230.00
SH2  2  2235.00
```

**7. 利用空值查询**

前面已经介绍了空值的概念，SQL 支持空值，可以利用空值进行查询。下面通过例子来查看 SQL 是如何利用空值来进行查询的。

**例 4.20** 找出尚未确定供应商的订购单。

```
SELECT * FROM 订购单 WHERE 供应商号 IS NULL
```

结果是

```
A6  NULL  OR77  NULL
A1  NULL  OR80  NULL
A3  NULL  OR90  NULL
```

**提示：**查询空值时要使用“IS NULL”，而“= NULL”是无效的，因为空值不是一个确定的值，所以不能用“=”这样的运算符进行比较。

**例 4.21** 列出已经确定了供应商的订购单信息。

```
SELECT * FROM 订购单 WHERE 供应商号 IS NOT NULL
```

结果是

```
A3  S7  OR67  2008/06/28
A1  S4  OR73  2008/07/28
A7  S4  OR76  2008/05/25
A3  S4  OR79  2008/06/13
A3  S3  OR91  2008/07/31
```



## 8. 别名与自然连接查询

在连接操作中,经常需要使用关系名作为前缀,有时这样做显得很麻烦。因此,SQL 允许在 **FORM** 短语中为关系名定义别名。定义别名的格式为

<关系名> <别名>

其功能是在 SQL 的 **FROM** 短语中为关系名定义别名。

**例 4.22** 为基于 4 个关系的连接查询设置关系名的别名。

```
SELECT  供应商名 FROM  供应商, 订购单, 职工, 仓库;
WHERE  地址 = "北京" AND 城市 = "北京";
      AND 供应商.供应商号 = 订购单.供应商号;
      AND 订购单.职工号 = 职工.职工号;
      AND 职工.仓库号 = 仓库.仓库号
```

结果是

北京奥运福娃制造厂  
北京奥运福娃制造厂  
北京爱民开关厂

**例 4.23** 使用别名的连接嵌套查询。

```
SELECT  供应商名 FROM  供应商 S, 订购单 P, 职工 A , 仓库 SH;
WHERE  地址 = "北京" AND 城市 = "北京";
      AND S.供应商号 = P.供应商号;
      AND P.职工号 = A.职工号;
      AND A.仓库号 = SH.仓库号
```

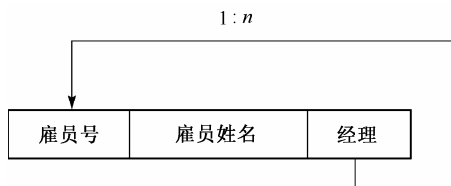
在以上例子中,别名不是必需的,但在关系的自然连接中,别名是必不可少的。

上面提到了自然连接,那么什么是自然连接?SQL 不仅可以对多个关系实行连接操作,也可以将同一个关系与其自身进行连接,这种连接就称为自然连接。在可以进行这种自然连接的关系中,实际存在着一种特殊的递归关系,即关系中的一些元组,根据出自同一值域的两个不同的属性,可以与另外的一些元组有一种对应关系(一对多联系)。

下面举个例子来说明自然连接,假设雇员关系(参见图 4.59):

雇员(雇员号, 雇员姓名, 经理)

其中雇员号和经理两个属性出自同一值域,同一元组的这两个属性值是上下级关系。



雇员号	雇员姓名	经理
A3	赵晓林	
A4	赵和平	A3
A6	王晓宏	A3
A8	王大勋	A6

图 4.59 雇员关系

例 4.24 根据雇员关系列出上一级经理及其所领导的职员名单。

```
SELECT S.雇员姓名, "领导", A.雇员姓名 FROM 雇员 S, 雇员 A;  
WHERE S.雇员号 = A.经理
```

结果是

赵晓林	领导	赵和平
赵晓林	领导	王晓宏
王晓宏	领导	王大勋

该例中通过别名形成了两个关系，一个是经理关系 S，一个是雇员关系 A，通过两个关系的连接实现了检索要求。在 SELECT 短语中也可以有常量，如该例中的“领导”。

9. 内外层互相关嵌套查询

在嵌套查询中，有时候内层查询的条件需要外层查询提供值，而外层查询的条件需要内层查询的结果，此时称为互相关。例如，在订购单中加入一个新字段“总金额”，说明完成该订购单所应付出的总金额数，新的订购单表的记录如表 4.2 所示。

表 4.2 订购单记录

职工号	供应商号	订购单号	订购日期	总金额
A3	S7	OR67	2004-6-28	45300
A1	S4	OR73	2004-7-28	23400
A7	S4	OR76	2004-5-25	8765
A6		OR77	- -	6450
A3	S4	OR79	2004-6-13	40005.45
A1		OR80	- -	35690
A3		OR90	- -	7869.5
A3	S3	OR91	2004-7-31	22540

例 4.25 列出每个职工经手的具有最高总金额的订购单信息。

```
SELECT out.职工号, out.供应商号, out.订购单号, out.订购日期, out.总金额;  
FROM 订购单 out WHERE 总金额=;  
(SELECT MAX(总金额) FROM 订购单 inner1;  
WHERE out.职工号=inner1.职工号)
```

结果是

A3	S7	OR67	2008/06/28	45300.00
A7	S4	OR76	2009/05/25	8765.00
A6		OR77		6450.00
A1		OR80		35690.00

在这个例子中，外层查询和内层查询使用同一个关系。外层查询提供 out 关系中每个元组的职工号值给内层查询使用；内层查询利用这个职工号值，确定该职工经手的具有最高总金额的订购单的总金额；随后外层查询再根据 out 关系的同一元组的总金额值与该总金额值进行比较，如果相等，则该元组被选择。

## 10. 有特殊符号的查询

### 1) 有特殊符号的查询

在 SQL-SELECT 中常会用到以下几个特殊的运算符：

- BETWEEN...AND...
- LIKE
- != 或 NOT

下面通过几个例子来理解这几个运算符的含义和用途。

**例 4.26** 检索出工资在 2220 元到 2240 元范围内的职工信息。

```
SELECT * FROM 职工 WHERE 工资 BETWEEN 2220 AND 2240
```

结果是

```
SH2 A1 2220  
SH3 A6 2230
```

这里的 BETWEEN...AND...的意思是“在...和...之间”，这个查询又等价于：

```
SELECT * FROM 职工 WHERE (工资 > 2220) AND (工资 < 2240)
```

可见使用 BETWEEN...AND...表达条件更清晰、更简洁。

**例4.27** 从供应商关系中检索出全部公司的信息，不要工厂或其他供应商的信息。因为这是一个匹配字符串的查询，应该使用 LIKE 运算符。

```
SELECT * FROM 供应商 WHERE 供应商名 LIKE "%电子厂"
```

LIKE 是字符串匹配运算符，有两种匹配符：

- 匹配符 “%”：表示 0 个或多个字符
- 匹配符 “\_”：表示一个字符

**例 4.28** 找出不在北京的全部供应商信息。

```
SELECT * FROM 供应商 WHERE 地址 != "北京"
```

结果是

```
S3 长安新悦电子厂 西安  
S6 河南新郑电子厂 郑州
```

在 SQL 中，用 “!=” 表示不等于。也可用否定运算符 NOT 写出等价命令：

```
SELECT * FROM 供应商 WHERE NOT( 地址 = "北京" )
```

NOT 的应用很广，比如 NOT IN、NOT BETWEEN 等。例如，找出工资不在 2220 元和 2240 元之间的全部职工信息的 SQL 命令为

```
SELECT * FROM 职工 WHERE 工资 NOT BETWEEN 2220 AND 2240
```

结果是

```
SH1 A3 2210
```

```
SH2  A4  2250
SH1  A7  2250
```

## 2) 使用量词和谓词的查询

使用量词和谓词的查询语句的格式为

<表达式> <比较运算符> [ANY|ALL|SOME] (子查询) [NOT] EXISTS (子查询)

式中 ANY、ALL、SOME 是量词，其中 ANY 和 SOME 是同义词。在进行比较运算时，只要子查询中有一行能使结果为真，则结果就为真；而 ALL 则要求子查询中的所有行都为真，结果才为真。EXISTS 是谓词，EXISTS 或 NOT EXISTS 是用来检查子查询中是否有结果返回，即存在元组或不存在元组。

**例 4.29** 检索哪些仓库中还没有职工的仓库的信息。

```
SELECT * FROM 仓库 WHERE NOT EXISTS;
      (SELECT * FROM 职工 WHERE 仓库号 = 仓库.仓库号)
```

**提示：**这里的内层查询引用了外层查询的表，只有这样使用谓词 EXISTS 或 NOT EXISTS 才有意义。所以这类查询都是内外层互相关的嵌套查询。

以上查询等价于

```
SELECT * FROM 仓库 WHERE 仓库号 NOT IN(SELECT 仓库号 FROM 职工)
```

结果是

```
SH4  武汉  1400
```

**例 4.30** 检索哪些仓库中至少已经有一个职工的仓库的信息。

```
SELECT * FROM 仓库 WHERE EXISTS;
      (SELECT * FROM 职工 WHERE 仓库号 = 仓库.仓库号)
```

以上查询等价于

```
SELECT * FROM 仓库 WHERE 仓库号 IN;
      (SELECT 仓库号 FROM 职工)
```

结果是

```
SH1  北京  1370
SH2  上海  1500
SH3  广州  1200
```

**例 4.31** 检索有职工的工资大于或等于 SH1 仓库中任何一名职工工资的仓库号（注意：这个查询使用 ANY 或 SOME 量词）。

```
SELECT DISTINCT 仓库号 FROM 职工 WHERE 工资 >= ANY;
      (SELECT 工资 FROM 职工 WHERE 仓库号 = "SH1")
```

以上查询等价于

```
SELECT DISTINCT 仓库号 FROM 职工 WHERE 工资 >=;
      (SELECT MIN(工资) FROM 职工 WHERE 仓库号 = "SH1")
```

结果是

```
SH1
SH2
SH3
```

**例4.32** 检索有职工的工资大于或等于 SH1 仓库中所有职工工资的仓库号 (注意: 这个查询使用 ALL 量词)。

```
SELECT DISTINCT 仓库号 FROM 职工 WHERE 工资 >= ALL;
(SELECT 工资 FROM 职工 WHERE 仓库号 = "SH1")
```

以上查询等价于

```
SELECT DISTINCT 仓库号 FROM 职工 WHERE 工资 >=;
(SELECT MAX(工资) FROM 职工 WHERE 仓库号 = "SH1")
```

结果是

```
SH1
SH2
```

## 11. 超连接查询

超连接查询首先保证一个表中满足条件的元组都在结果表中, 然后将满足连接条件的元组与另一个表的元组进行连接, 不满足连接条件的则应将来自另一表的属性值置为空值。

在一般的 SQL 语句中, 超连接运算符包括以下几种。

① “\*=” 左连接, 含义是在结果表中包含第一个表中满足条件的所有记录; 如果有在连接条件上匹配的元组, 则第二个表返回相应的值, 否则返回空值。

② “=\*” 右连接, 含义是在结果表中包含第二个表中满足条件的所有记录; 如果有在连接条件上匹配的元组, 则第一个表返回相应的值, 否则返回空值。

**提示:** Visual FoxPro 不支持超连接运算符 “\*=” 和 “=\*”, Visual FoxPro 使用专门的连接运算符语法格式来支持超连接查询。

超连接查询的格式为

```
SELECT ...
FROM Table INNER|LEFT|RIGHT|FULL JOIN Table
ON JoinCondition
WHERE ...
```

其中:

- INNER JOIN 等价于 JOIN, 为普通的连接, 在 VFP 中称为内部连接。
- LEFT JOIN 称为左连接。
- RIGHT JOIN 称为右连接。
- FULL JOIN 称为全连接, 即两个表中的记录不论是否满足连接条件都将在目标表或查询结果中出现, 不满足连接条件的记录对应部分为 NULL。
- ON JoinCondition 指定连接条件。

**例 4.33** 内部连接，即只有满足连接条件的记录才出现在查询结果中。

```
SELECT 仓库.仓库号, 城市, 面积, 职工号, 工资;  
FROM 仓库 JOIN 职工;  
ON 仓库.仓库号 = 职工.仓库号
```

如下两种命令格式也是等价的:

```
SELECT 仓库.仓库号, 城市, 面积, 职工号, 工资;  
FROM 仓库 INNER JOIN 职工;  
ON 仓库.仓库号 = 职工.仓库号
```

和

```
SELECT 仓库.仓库号, 城市, 面积, 职工号, 工资;  
FROM 仓库, 职工 WHERE 仓库.仓库号 = 职工.仓库号
```

结果是

```
SH2 上海 1500 A1 2220  
SH1 北京 1370 A3 2210  
SH2 上海 1500 A4 2250  
SH3 广州 1200 A6 2230  
SH1 北京 1370 A7 2250
```

**例4.34** 左连接，即满足连接条件的记录出现在查询结果中，第一个表中不满足连接条件的记录也出现在查询结果中。

```
SELECT 仓库.仓库号, 城市, 面积, 职工号, 工资;  
FROM 仓库 LEFT JOIN 职工;  
ON 仓库.仓库号 = 职工.仓库号
```

结果是

```
SH1 北京 1370 A3 2210  
SH1 北京 1370 A7 2250  
SH2 上海 1500 A1 2220  
SH2 上海 1500 A4 2250  
SH3 广州 1200 A6 2230  
SH4 武汉 1400 NULL NULL
```

**例4.35** 右连接，即满足连接条件的记录出现在查询结果中，第二个表中不满足连接条件的记录也出现在查询结果中。

```
SELECT 仓库.仓库号, 城市, 面积, 职工号, 工资;  
FROM 仓库 RIGHT JOIN 职工;  
ON 仓库.仓库号 = 职工.仓库号
```

结果是

```
SH2 上海 1500 A1 2220  
SH1 北京 1370 A3 2210  
SH2 上海 1500 A4 2250  
SH3 广州 1200 A6 2230  
SH1 北京 1370 A7 2250
```

**例4.36** 全连接，即满足连接条件的记录出现在查询结果中，两个表中不满足连接条件的记录也出现在查询结果中。

```
SELECT  仓库.仓库号, 城市, 面积, 职工号, 工资;
FROM    仓库 FULL JOIN 职工;
ON      仓库.仓库号 = 职工.仓库号
```

结果是

```
SH1  北京  1370  A3   2210
SH1  北京  1370  A7   2250
SH2  上海  1500  A1   2220
SH2  上海  1500  A4   2250
SH3  广州  1200  A6   2230
SH4  武汉  1400  NULL NULL
```

Visual FoxPro 的 SQL SELECT 语句的连接格式只能实现两个表的连接, 如果要实现多个表的连接, 还需要使用标准格式:

```
SELECT  仓库.仓库号, 城市, 供应商名, 地址;
FROM    供应商, 订购单, 职工, 仓库;
WHERE   供应商.供应商号 = 订购单.供应商号;
        AND 订购单.职工号 = 职工.职工号;
        AND 仓库.仓库号 = 职工.仓库号
```

**提示:** 在 JOIN 连接格式中, JOIN 的顺序和 ON 的顺序很重要, 特别要注意 JOIN 的顺序和 ON 的顺序 (相应的连接条件) 正好相反, 例如:

```
SELECT  仓库.仓库号, 城市, 供应商名, 地址;
FROM    供应商 JOIN 订购单 JOIN 职工 JOIN 仓库;
ON      职工.仓库号 = 仓库.仓库号;
ON      订购单.职工号 = 职工.职工号;
ON      供应商.供应商号 = 订购单.供应商号
```

## 12. 集合的并运算

SQL 支持集合的并 (UNION) 运算, 即将两个 SELECT 语句的查询结果通过并运算合并成一个查询结果。为了进行并运算, 要求这样的查询结果具有相同的字段个数, 并且对应字段的值要出自同一个值域, 即具有相同的数据类型和取值范围。

**例 4.37** 查询北京和上海的仓库信息。

```
SELECT  * FROM 仓库 WHERE 城市 = "北京";
UNION;
SELECT  * FROM 仓库 WHERE 城市 = "上海"
```

## 13. SQL SELECT 中的几个特殊选项

### 1) 显示部分结果

在应用中有时只需要满足条件的前几个记录, 这时使用 “TOP nExpr[PERCNT]” 语句非常有用。其中, nExpr 是数字表达式, 当不使用 PERCNT 时, nExpr 是 1~32 767 之间的整数,

表示显示前几个记录；当使用 PERCNT 时，nExpr 是 0.01~99.99 之间的实数，表示显示前百分之几的记录。

**例 4.38** 显示工资最高的前 3 位职工的信息。

```
SELECT * TOP 3 FROM 职工 ORDER BY 工资 DESC
```

结果是

```
SH2  A4  2250
SH1  A7  2250
SH3  A6  2230
```

**例 4.39** 显示工资最低的 30% 职工的信息。

```
SELECT * TOP 30 PERCENT FROM 职工 ORDER BY 工资
```

结果是

```
SH1  A3  2210
SH2  A1  2220
```

**提示：**TOP 短语要与 ORDER BY 短语同时使用才有效。

## 2) 将查询结果放在数组中

使用“**INTO ARRAY ArrayName**”语句将查询结果放在数组中。ArrayName 可以是任意数组变量名。例如，将查询到的职工信息存放在数组 SZ 中：

```
SELECT * FROM 职工 INTO ARRAY SZ
```

## 3) 将查询结果存放在临时文件中

使用“**INTO CURSOR CursorName**”语句将查询结果放在临时数据库表文件中。该表为只读的.dbf 文件，当关闭文件时该文件将自动删除。例如，将查询到的职工信息存放在临时表文件 ZGB 中：

```
SELECT * FROM 职工 INTO CURSOR ZGB
```

## 4) 将查询结果存放在永久表中

使用“**INTO DBF|TABLE TableName**”语句将查询结果放在永久表中（.dbf 文件）。例如，将例 4.38 的查询结果存放在表文件 ZGB1 中：

```
SELECT * TOP 3 FROM 职工 INTO TABLE ZGB1 ORDER BY 工资 DESC
```

## 5) 将查询结果存放在文本文件中

使用“**TO FILE FileName[ADDITIVE]**”语句将查询结果放在文本文件中。其中，ADDITIVE 选项使结果追加到原有文件的尾部，否则将覆盖原有文件。例如，将例 4.38 的查询结果以文本的形式存放在文本文件 WBWJ 中：

```
SELECT * TOP 3 FROM 职工 TO FILE WBWJ ORDER BY 工资 DESC
```

如果 TO 短语和 INTO 短语同时使用，则 TO 短语将会被忽略。

## 6) 将查询结果直接输出到打印机

使用“**TO PRINTER [PROMPT]**”语句将查询结果直接输出到打印机。如果使用 PROMPT 选项，在开始打印之前打开打印机设置对话框。



### 4.3.3 操作功能

SQL 的操作功能是指对数据库中数据的操作，包括数据的插入、数据的更新和数据的删除。下面分别介绍。

#### 1. 数据插入功能

VFP 支持两种插入命令格式，一种是标准格式，另一种是特殊格式。

##### ● 标准格式

```
INSERT INTO dbf_name[( fname1[, fname2...])]
VALUES (eExpression1[, eExpression2,...])
```

##### ● 特殊格式

```
INSERT INTO dbf_name FROM ARRAY ArrayName | FROM MEMVAR
```

需要说明的是

① INSERT INTO dbf\_name 说明向由 dbf\_name 指定的表中插入记录，当插入的不是完整的记录时，可以用 fname1, fname2...指定字段。

② VALUES(eExpression1[, eExpression2,...])给出具体的记录值。

③ FROM ARRAY ArrayName 说明从指定的数组中插入记录值。

④ FROM MEMVAR 说明根据同名的内存变量来插入记录值，如果同名变量不存在，那么相应的字段为默认值或空。

**例 4.40** 向订购单关系中插入元组: ("A7", "S4", "OR01", 2004/05/25)。

```
INSERT INTO 订购单 VALUES ("A7", "S4", "OR01", {^2004-05-25})
```

假如供应商未确定，只能先插入职工号和订购单号两个属性的值。

```
INSERT INTO 订购单(职工号, 订购单号) VALUES ("A7", , "OR01")
```

这时，另外两个属性值为空。

下列一组命令说明 INSERT INTO FROM ARRAY 语句的使用方法：

- USE 订购单     && 打开订购单
- SCATTER TO arrA     && 将当前记录读到数组 arrA
- COPY STRUCTURE TO AHG     && 将订购单表结构复制到 AHG
- INSERT INTO AHG FROM ARRAY arrA     && 从数组 arrA 插入一条记录到 AHG
- SELECT AHG     && 切换 AHG 的工作区
- BROWSE     && 显示插入记录的结果
- DELETE FILE AHG.dbf     && 关闭并删除 AHG.dbf 表

下列一组命令说明“INSERT INTO FROM MEMVAR”的使用方法：

- USE 订购单
- SCATTER MEMVAR     && 将当前记录读到内存变量

- COPY STRUCTURE TO AHG
- INSERT INTO AHG FROM MEMVAR    && 从内层变量插入一条记录到 AHG
- SELECT AHG
- BROWSE
- USE
- DELETE FILE AHG.dbf

**提示：**当一个表定义了主索引或候选索引后，由于相应的字段具有关键字的特性，即不可能为空，所以只能用此命令插入记录。Visual FoxPro 以前的插入命令（INSERT 或 APPEND）是先插入一条空记录，然后再输入各字段的值。由于关键字字段不允许为空，所以使用以前的方法不能成功插入记录。

## 2. 数据更新功能

SQL 的数据更新命令的格式如下：

```
UPDATE TableName
SET Column_Name1 = eExpresssion1, [ , Column_Name2=eExpresssion2...]
[WHERE Condition]
```

例如，给 SH1 仓库的职工提高 10% 的工资：

```
UPDATE 职工 SET 工资 = 工资*1.10 WHERE 仓库号 = "SH1"
```

## 3. 数据删除功能

SQL 的数据删除命令的格式如下：

```
DELETE FROM TableName [WHERE Condition]
```

例如，要删除仓库关系中仓库号值是 SH2 的元组：

```
DELETE FROM 仓库 WHERE 仓库号 = "SH2"
```

此删除同样是逻辑删除记录。

### 4.3.4 定义功能

标准的 SQL 的数据定义功能包括数据库的定义、表的定义、视图的定义、存储过程的定义、规则的定义和索引的定义。Visual FoxPro 支持表的定义和视图的定义。

#### 1. 表的定义

除了通过表设计器建立表的方法之外，在 Visual FoxPro 中也可以通过 SQL 的 CREATE TABLE 命令建立表，相应的命令格式是

```
CREATE TABLE <表名>
(<列名> <数据类型> [列级完整性约束条件]
...
[ , <表级完整性约束条件> ] )
```

从以上句法格式中可以看出,使用 **CREATE TABLE** 命令建立表可以完成用表设计器实现的所有功能。除了建立表的基本功能之外,它还包括满足实体完整性的主关键字(主索引) **PRIMARY KEY**、定义域完整性的 **CHECK** 约束及出错提示信息 **ERROR**、定义默认值的 **DEFAULT** 等。另外还有描述表之间联系的 **FOREIGN KEY** 和 **REFERENCES** 等。

**CREATE TABLE** 命令中可以使用的数据类型及说明在第3章已经介绍过,这里不再重复。

可以通过数据库设计器和表设计器建立数据库,也可以利用 **SQL** 命令来建立相同的数据库。然后就能利用数据库设计器和表设计器来检验用 **SQL** 建立的数据库。

#### 例 4.41 使用下列命令建立“订货管理”数据库。

```
CREATE DATABASE 订货管理
```

使用 **CREATA** 命令建立仓库 A 表:

```
CREATE TABLE 仓库 A(仓库号 C(5) PRIMARY KEY, 城市 C(10),;  
面积 I CHECK(面积>0) ERROR "面积应该大于 0!")
```

这里使用的 **TABLE** 和 **DBF** 是等价的,前者是标准 **SQL** 的关键词,后者是 **Visual FoxPro** 的关键词;以上命令在当前打开的“订货管理”数据库中建立了仓库 A 表,其中仓库号是主关键字(主索引,用 **PRIMARY KEY** 说明),使用 **CHECK** 为面积字段值说明了有效性规则(面积>0),使用 **ERROR** 为该有效性规则说明了出错提示信息“面积应该大于 0!”。如果“订货管理”数据库设计器没有打开,可以使用 **MODIFY DATABASE** 命令打开,那么执行完如上命令后在数据库设计器中立刻可以看到该表。

#### 例 4.42 使用 **CREATE** 命令建立职工 A 表。

```
CREATE TABLE 职工 A(仓库号 C(5), 职工号 C(5) PRIMARY KEY, 工资  
I CHECK(工资 >= 1000 AND 工资 <= 5000)  
ERROR "工资值的范围在 1000-5000!" DEFAULT 2200,  
FOREIGN KEY 仓库号 TAG 仓库号 REFERENCES 仓库 A)
```

#### 例 4.43 使用 **CREATE** 命令建立供应商 A 表。

```
CREATE TABLE 供应商 A(  
供应商号 C(5) PRIMARY KEY, 供应商名 C(20), 地址 C(20))
```

使用 **CREATE** 命令建立订购单 A 表。

```
CREATE TABLE 订购单 A  
职工号 C(5),  
供应商号 C(5),  
订购单号 C(5) PRIMARY KEY,  
订购日期 D,  
FOREIGN KEY 职工号 REFERENCES 职工 A,  
FOREIGN KEY 供应商号 TAG 供应商号 REFERENCES 供应商 A)
```

在上面的示例中,使用了 **CREATE** 命令的主要格式,还有一些具有 **Visual FoxPro** 特色的关键词或短语,如下所示:

- **NAME LongTableName**: 为建立的表指定一个长名。

- FREE: 建立的表不添加到当前数据库中, 即建立一个自由表。
- NULL 或 NOT NULL: 说明字段允许或不允许为空值。
- UNIQUE: 说明建立候选索引 (注意不是唯一索引)。
- FROM ARRAY ArrayName: 说明根据指定数组的内容建立表, 数组的元素依次是字段名、类型等。建议不使用此方法。

提示: ① 使用 SQL CREATE 命令新建的表自动在最低可用工作区打开, 并可以通过别名引用, 新表的打开方式为独占方式, 忽略 SET EXCLUSIVE 的当前设置。  
② 如果建立自由表 (当前没有打开的数据库或使用了 FREE), 则很多选项在命令中不能使用, 例如 NAME、CHECK、DEFAULT、FOREIGN KEY、PRIMARY KEY 和 REFERENCES 等。

## 2. 表的删除

删除表的 SQL 命令格式是

```
DROP TABLE table_name
```

DROP TABLE 命令直接从磁盘上删除 table\_name 所对应的.dbf 文件。如果 table\_name 是数据库中的表并且相应的数据库是当前数据库, 则从数据库中删除该表。否则, 虽然从磁盘上删除了.dbf 文件, 但是记录在数据库.dbf 文件中的信息却没有删除, 此后还会出现错误提示。所以要删除数据库中的表时, 最好应使数据库是当前打开的数据库, 并在数据库中进行操作。

例 4.44 使用 SQL 删除 Student 表。

```
DROP TABLE Student
```

提示: 基本表定义一旦删除, 表中的数据、此表上建立的索引和视图都将自动删除。

## 3. 修改表结构

SQL 中修改表结构的命令是 “ALTER TABLE”, 该命令有三种格式。

### ① 格式 1

```
ALTER TABLE <表名>  
ADD | ALTER [COLUMN] <字段名> <类型> [( <宽度>[,<小数>])  
[NULL|NOT NULL]  
[CHECK <逻辑表达式>[ERROR <字符表达式>]]  
[DEFAULT <表达式>]  
[PRIMARY KEY | UNIQUE]  
[REFERENCES <表名 2> [TAG <标识>]]  
[NOCPTRANS]
```

该格式可以添加 (ADD) 新的字段或修改 (ALTER) 已有的字段, 它的句法基本可以与 CREATE TABLE 的句法相对应。

例如, 为订购单 A 表增加一个货币类型的总金额字段:

```
ALTER TABLE 订购单 A;
      ADD 总金额 Y CHECK 总金额 > 0 ERROR "总金额应该大于 0!"
```

或者将订购单 A 表的订购单号字段的宽度由原来的 5 改为 6:

```
ALTER TABLE 订购单 1 ALTER 订购单号 C(6)
```

从以上命令格式可以看出, 该格式可以修改字段的类型、宽度、有效性规则、错误性信息、默认值、定义主关键字和关联等; 但是不能修改字段名, 不能删除字段, 也不能删除已经定义的规则等。

## ② 格式 2

```
ALTER TABLE <表名>
ALTER [COLUMN] <字段名>
      [NULL | NOT NULL]
      [SET DEFAULT <表达式>]
      [SET CHECK <逻辑表达式>[ERROR < 字符表达式>]]
      [DROP DEFAULT]
      [DROP CHECK]
```

从命令格式可以看出, 该格式主要用于定义、修改和删除有效性规则及默认值定义。

例如, 修改或定义总金额字段的有效性规则:

```
ALTER TABLE 订购单 A
      ALTER 总金额 SET CHECK 总金额 > 100 ;
      ERROR "总金额应该大于 100"
```

还有删除总金额字段的有效性规则:

```
ALTER TABLE 订购单 1 ALTER 总金额 DROP CHECK
```

以上两种格式都不能删除字段, 也不能更改字段名, 所有修改是在字段一级。第三种格式是在这些方面对前两种格式的补充。

## ③ 格式 3

```
ALTER TABLE <表名 1>
      [DROP [COLUMN] <字段名 1>]
      [SET CHECK <逻辑表达式 1>[ERROR <字符表达式>]]
      [DROP CHECK]
      [ADD PRIMARY KEY <表达式 1> TAG <标识名 1> [FOR <逻辑表达式 2>]]
      [DROP PRIMARY KEY]
      [ADD UNIQUE <表达式 2> TAG <标识名 2> FOR <逻辑表达式 3>]
      [DROP UNIQUE TAG <标识名 3>]
      [ADD FOREIGN KEY <表达式 3> TAG <标识名 4> [FOR <逻辑表达式 4>]
      REFERENCES <表名 2> [TAG <标识名 5>]]
      [DROP FOREIGN KEY TAG <标识名 6> [SAVE]]
      [RENAME COLUMN <字段名 2> TO <字段名 3>]
      [NOVALIDATE]
```

该格式可以删除字段（DROP [COLUMN]），可以修改字段名（RENAME COLUMN），也可以定义、修改和删除表一级的有效性规则等。

例如，将订购单 A 表的“总金额”字段名改为“金额”：

```
ALTER TABLE 订购单 A RENAME COLUMN 总金额 TO 金额
```

删除订购单 A 表中的“金额”字段：

```
ALTER TABLE 订购单 A DROP COLUMN 金额
```

将订购单 A 表的职工号和供应商号定义为候选索引（候选关键字），索引名是 XP：

```
ALTER TABLE 订购单 A ADD UNIQUE 职工号+供应商号 TAG XP
```

或是删除订购单 A 表的候选索引 XP：

```
ALTER TABLE 订购单 A DROP UNIQUE TAG XP
```

### 4.3.5 定义视图

在 Visual FoxPro 中，视图是一个定制的虚表，可以是本地的、远程的或带参数的。视图可引用一个或多个表，或者引用其他视图。视图是可更新的，它可以引用远程表。

在关系数据库中，视图与窗口有特殊含义。视图也称为窗口，即视图是操作表的窗口，可以把它看成是从表中派生出来的虚表。它依赖于表，但不能独立存在。

由于视图是从一个或几个基本表（或视图）导出的表，因此与基本表不同，是一个虚表。数据库中只存放视图的定义，而不存放视图对应的数据，这些数据仍存放在原来的基本表中。所以基本表中的数据发生变化，从视图中查询出的数据也就随之改变了。

视图一经定义，就可以和基本表一样被查询、删除。可以在一个视图之上再定义新的视图，但对视图的更新（增加、删除、修改）操作则有一定的限制。

视图是根据对表的查询定义的，其命令格式如下：

```
CREATE VIEW <视图名> [( <列名>[, <列名>]...)]  
AS<子查询>  
[WITH CHECK OPTION];
```

需要注意以下几点。

① 子查询可以是任意复杂的 SELECT 语句，但通常不允许含有 ORDER BY 子句和 DISTINCT 短语。

② WITH CHECK OPTION 表示对视图进行 UPDATE、INSERT 和 DELETE 操作时要保证更新、插入或删除的行满足视图定义中的谓词条件（即子查询中的条件表达式）。

③ 组成视图的属性列名或者全部省略或者全部指定，没有第 3 种选择。如果省略了视图的各个属性列名，则表示该视图由子查询的 SELECT 子句目标列中的诸字段组成。但在下列 3 种情况中必须明确指定组成视图的所有列名：

- 某个目标列不是单纯的属性名，而是函数或列表表达式。
- 多表连接时选出了几个同名列作为视图的字段。
- 需要在视图中为某个列启用新的更合适的名字。

## 1. 从单个表派生出的视图

如果某个用户对职工关系只需要或者只能知道职工号和所工作的仓库号，那么可以定义视图：

```
CREATE VIEW 视图1 AS;  
SELECT 职工号, 仓库号 FROM 职工
```

其中“视图1”是视图的名称。视图一经定义，就可以和基本表一样进行各种查询，也可以进行一些修改操作。对于最终用户来说，有时并不需要知道操作的是基本表还是视图。

为了查询职工号和仓库号信息，可以给出如下命令：

```
SELECT * FROM 视图1  
或 SELECT 职工号, 仓库号 FROM 视图1  
或 SELECT 职工号, 仓库号 FROM 职工
```

上面的语句限定了列构成的视图，下面再限定行定义的一个视图。例如，某个用户对仓库关系只需要或者只能查询北京仓库的信息，可以定义如下视图：

```
CREATE VIEW 视图2 AS;  
SELECT 仓库号, 面积 FROM 仓库 WHERE 城市 = "北京"
```

这里“视图2”中只有北京仓库的信息，所以城市属性就不再需要。

**提示：**视图一方面可以限定对数据的访问，另一方面又可以简化对数据的访问。

## 2. 从多个表派生出的视图

有些查询是很复杂的，比如例4.25中列出每个职工经手的具有最高总金额的订购单信息，可以使用视图来完成。

**例 4.45** 列出每个职工经手的具有最高总金额的订购单信息。

```
CREATE VIEW 视图3 AS;  
SELECT out.职工号, out.供应商号, out.订购单号, out.订购日期, out.总金额;  
FROM 订购单 out WHERE 总金额 =;  
(SELECT MAX(总金额) FROM 订购单 inner1;  
WHERE out.职工号 = inner1.职工号)
```

这时候再提出同样的查询要求，就只需要输入以下命令即可：

```
SELECT * FROM 视图3
```

**例 4.46** 建立视图向用户提供职工号、职工的工资和职工工作所在城市的信息。

```
CREATE VIEW 视图4 AS;  
SELECT 职工号, 工资, 城市 FROM 职工, 仓库;  
WHERE 职工.仓库号 = 仓库.仓库号
```

结果对于用户就好像有一个包含字段“职工号”、“工资”和“城市”的表。

### 3. 视图中的虚字段

利用查询来建立一个视图的 `SELECT` 子句可以包含算术表达式或函数，可以作为视图的字段。由于它们是计算得来的，并不存储在表内，所以称为虚字段。

**例 4.47** 定义一个视图，其中包含职工号、月工资和年工资 3 个字段。

```
CREATE VIEW 视图 5 AS;  
SELECT 职工号, 工资 AS 月工资, 工资*10 AS 年工资 FROM 职工
```

这里在 `SELECT` 短语中利用 `AS` 重新定义了视图的字段名。由于其中有一个字段是计算得来的，所以必须给出字段名。这里年工资是虚字段，它是由职工表的工资字段乘以 12 得到的；而月工资就是职工表中的工资字段，由此可见在视图中还可以重新命名字段名。

查询视图 5 的命令如下：

```
SELECT * FROM 视图 5
```

结果是

```
A1  2220  22200  
A3  2210  22100  
A4  2250  22500  
A6  2230  22300  
A7  2250  22500
```

### 4. 删除视图

由于视图是从表中派生出来的，所以不存在修改结构的问题，但是视图可以删除。在 VFP 中可以修改视图，将在第 5 章介绍。删除视图的命令格式是

```
DROP VIEW <视图名>
```

例如，使用 SQL 命令删除“视图 2”：

```
DROP VIEW 视图 2
```

### 5. 查询视图

定义视图后，用户就可以像对基本表一样对视图进行查询了。

**例 4.48** 在视图中找出学时大于 50 的教师信息。

```
SELECT 教师编号, 教师姓名, 课程编号 FROM 视图 WHERE 学时 > 50
```

在数据库管理系统中，当执行对视图的查询时，首先进行有效性检查，检查查询的表、视图等是否存在。如果存在，则从数据字典中取出视图的定义，把定义中的子查询和用户的查询结合起来，转换成等价的对基本表的查询，然后再执行修正后的查询。这一转换过程称为视图消解。

**例 4.49** 查询计算机系选修了 CS02 课程的学生。

```
SELECT 学号, 姓名 FROM 学生成绩, 学生选课;  
WHERE 学生成绩.学号 = 学生选课.学号 AND 学生选课.课程编号 = "CS02"
```

本查询涉及虚表“学生成绩”和基本表“学生选课表”，通过这两个表的连接来完成用户请求。



在一般情况下,视图查询的转换是直接的。但有些情况下,这种转换不能直接进行,查询时就会出现问题,如下所示。

**例 4.50** 在“成绩表”视图中查询平均成绩在 90 分以上的学生学号和总评成绩。

```
SELECT * FROM 成绩表 WHERE 总评成绩 >= 90
```

“学生成绩”视图定义为

```
SELECT 学号, AVG(总评成绩);  
FROM 成绩表;  
GROUP BY 学号
```

将上面查询语句与子查询结合。形成下列查询语句:

```
SELECT 学号, AVG(总评成绩);  
FROM 成绩表;  
GROUP BY 学号;  
HAVING AVG(总评成绩) >= 90
```

目前,多数关系数据库系统对行列子集视图的查询均能进行正确转换。但对非行列子集的查询就不一定能做转换了,因此这类查询应该直接对基本表进行。

## 6. 更新视图

更新视图是指通过视图来插入 (INSERT)、删除 (DELETE) 和修改 (UPDATE) 数据。由于视图是并不实际存储数据的虚表,因此对视图的更新最终要转换为对基本表的更新。

为防止用户通过视图对数据进行增加、删除、修改时,有意无意地对不属于视图范围内的基本表数据进行操作,可在定义视图时加上 **WITH CHECK OPTION** 子句。这样在视图上增、删、改数据时,系统会检查视图定义中的条件,若不满足条件,则拒绝执行该操作。

**例 4.51** 将“学生成绩”视图中学号为“074202123”的学生姓名改为“齐小平”。

```
UPDATE 学生成绩  
SET 姓名 = "齐小平";  
WHERE 学号 = "074202123"
```

**例 4.52** 向“学生成绩”视图中插入一个新的学生记录,其中学号为“074101069”,姓名为“赵鑫鑫”。

```
INSERT  
INTO 学生成绩;  
VALUES("074101069", "赵鑫鑫", "女", "CC03", "VFP 数据库程序设计", 87)
```

转换为对基本表的更新:

```
INSERT  
INTO 学生成绩(学号, 姓名, 性别, 课程编号, 课程名称, 成绩)  
VALUES("074101069", "赵鑫鑫", "女", "CC03", "VFP 数据库程序设计", 87)
```

系统自动将系名“IS”放入 VALUES 子句中。

一般情况下，行列子集视图是可更新的。除行列子集视图外，还有些视图在理论上是可更新的，但它们的确切特征还是尚待研究的课题。还有些视图在理论上是不可更新的。

## 7. 视图的作用

### 1) 视图能够简化用户的操作

视图机制使用户可以将注意力集中在所关心的数据上。如果这些数据不是直接来自基本表，则可以通过定义视图，使数据库看起来结构简单、清晰，并且可以简化用户的数据查询操作。例如，那些定义了若干张表连接的视图，就将表与表之间的连接操作对用户隐藏起来。换句话说，用户所做的只是对一个虚表的简单查询，而这个虚表是怎样得来的，用户无须了解。

### 2) 视图使用户能以多种角度看待同一数据

视图机制能使不同的用户以不同的方式看待同一数据。

### 3) 视图对重构数据库提供了一定程度的逻辑独立性

在关系数据库中，数据库的重构往往是不可避免的。重构数据库最常见的是将一个基本表“垂直”地分成多个基本表。

在关系数据库中，视图始终不是真正地含有数据，它总是原来表的一个窗口。所以，虽然视图可以像表一样进行各种查询，但是插入、更新和删除操作在视图上却有一定限制。在一般情况下，当一个视图是由单个表导出时可以进行插入和更新操作，但不能进行删除操作；当视图是从多个表导出时，插入、更新和删除操作都不允许进行。这种限制是很有必要的，它可以避免一些潜在问题的发生。

## 思考与练习 4

### 一、思考题

1. 查询和视图有何不同？各有何用途？
2. 如何设置查询的输出形式？
3. 如何创建参数化视图？
4. 超连接查询中的运算符有哪些？各有什么含义？
5. 在 SQL 中，删除视图用什么命令？

### 二、选择题

1. 视图设计器中比查询设计器中多出的选项卡是\_\_\_\_\_。  
A. 字段                      B. 排序依据                      C. 连接                      D. 更新条件
2. 在【添加表和视图】窗口中，【其他】按钮的作用是为了让用户选择\_\_\_\_\_。  
A. 数据库表                      B. 视图                      C. 不属于数据库的表                      D. 查询
3. 下列说法中，错误的是\_\_\_\_\_。  
A. 在数据库中，可以包含表、视图、查询及表间的永久关系。  
B. 可以通过修改视图中的数据来更新数据源中数据，但查询不可以。  
C. 查询和视图都是用 SELECT-SQL 语言实现的，都要以数据表作为数据源。  
D. 视图虽然具备了一般数据表的特征，但它本身并不是表。

4. 要在浏览窗口中显示表“js.dbf”中所有的“教授”和“副教授”记录, 下列命令中错误的是\_\_\_\_\_。
- A. USE JS BROWSE FOR 职称 = "教授" AND 职称 = "副教授"
- B. SELECT \* FROM JS WHERE "教授"\$ 职称
- C. SELECT \* FROM JS WHERE 职称 IN ("教授", "副教授")
- D. SELECT \* FROM JS WHERE LIKE ("\*教授", 职称)
5. SQL 语句中修改表结构的命令是\_\_\_\_\_。
- A. MODIFY TABLE B. MODIFY STRUCTURE
- C. ALTER TABLE D. ALTER STRUCTURE
6. 创建一个参数化视图时, 应在筛选对话框的实例框中输入\_\_\_\_\_。
- A. “\*”及参数名 B. “?”及参数名 C. “!”及参数名 D. 参数名
7. 当两张表进行无条件连接时, 交叉组合后形成的新记录个数是\_\_\_\_\_。
- A. 两张表记录数之差 B. 两张表记录数之和
- C. 两张表中记录多的记录数 D. 两张表记录数的乘积
8. 查询文件中保存的是\_\_\_\_\_。
- A. 查询的命令 B. 查询的结果 C. 查询的基表 D. 查询的条件
9. 有关查询设计器, 正确的描述是\_\_\_\_\_。
- A. 【连接】选项卡与 SQL 语句的 GROUP BY 短语对应
- B. 【筛选】选项卡与 SQL 语句的 HAVING 短语对应
- C. 【排序依据】选项卡与 SQL 语句的 ORDER BY 短语对应
- D. 【分组依据】选项卡与 SQL 语句的 JOIN ON 短语对应
10. 使用 SQL 语句将学生表 S 中年龄 (AGE) 大于 30 岁的记录删除, 正确的命令是\_\_\_\_\_。
- A. DELETE FOR AGE > 30 B. DELETE FROM S WHERE AGE > 30
- C. DELETE S FOR AGE > 30 D. DELETE S WHERE AGE > 30

### 三、填空题

1. 查询中的筛选条件可以通过 SQL 的 SELECT 语句的\_\_\_\_\_子句来实现。
2. 按课程和系名统计人数和平均成绩, 必须以\_\_\_\_\_作为分组的依据。
3. 交叉表查询建立好后, 可以在\_\_\_\_\_中打开并修改它。
4. 在 SQL 的 SELECT 语句中, 表示条件表达式用 WHERE 子句, 分组用\_\_\_\_\_子句, 排序用\_\_\_\_\_子句。
5. 在 SQL 的 SELECT 语句中用于计算检索的函数有 COUNT、\_\_\_\_、\_\_\_\_、MAX 和 MIN。
6. 在视图设计器中, 比查询设计器多出的选项卡是\_\_\_\_\_。
7. 在查询设计器中, 选择查询结果中出现的字段及表达式应在\_\_\_\_\_选项卡中完成, 设置查询条件应在\_\_\_\_\_选项卡中完成, 该选项卡相当于 SQL-SELECT 语句中的 WHERE 子句。
8. 在 Visual FoxPro 中, 使用 SQL 的 CREATE TABLE 语句建立数据库表, 使用\_\_\_\_\_子句说明有效性规则 (域完整性或字段取值范围)。
9. 在使用 SQL 的 SELECT 语句进行分组计算查询时, 可以使用\_\_\_\_\_子句来去掉不满足条件的分组。
10. 设有 s (学号, 姓名, 性别) 和 sc (学号, 课程号, 成绩) 两个表, 下面的 SELECT 语句检索选修的每门课程的成绩都高于或等于 85 分的学生的学号、姓名和性别:
- ```
SELECT 学号, 姓名, 性别 FROM S;  
WHERE _____(SELECT * FROM SC WHERE SC.学号 = S.学号 AND 成绩 < 85)
```

## 第 5 章 Visual FoxPro 程序设计基础

前几章我们经常会在【命令】窗口中输入命令来完成对数据库的各种操作。这种方法虽然简单，但是当数据很多且需要经常反复使用某些命令时，就显得非常繁琐。Visual FoxPro 将过程化程序设计与面向对象程序设计结合在一起，使用户可以创建出功能强大、灵活多变的应用程序。这一章主要介绍 Visual FoxPro 程序设计语言的基本语法知识。

### 5.1 程序设计基础

在 Visual FoxPro 系统中，用户可以通过菜单方式对数据库及数据表进行管理，不需要记住命令，操作比较方便。也可以使用交互式操作方法，但这种方式对于解决简单的问题和数据管理是方便的，但对于复杂的问题和大量重复的命令，利用这种方式进行处理是难以完成的。例如求 1~5000 之间的素数，如果用前面所学的知识或简单的几个命令，是很难实现的。为此，需要进行程序设计，Visual FoxPro 提供了程序执行方式来解决该问题。

Visual FoxPro 允许用户将一系列的语句或命令存储在一个文件中，形成一个程序文件（.prg）。用户只要执行该程序文件，Visual FoxPro 就会自动执行程序文件中的所有命令，这就是所谓的程序执行方式。因此，在 Visual FoxPro 中，不仅能够使用交互式执行方式操作数据库与数据表，而且可以通过运行程序文件，完成某些特殊的功能。从而实现通过程序控制自动执行一系列操作。程序操作方式不仅运行效率高，而且可重复执行。另一个好处是，虽然编程的人员需要熟悉 Visual FoxPro 的命令和掌握编程的方法，但使用程序的人员却只需了解程序的运行步骤和运行过程中的人机交互要求，而不必知道程序的内部结构和其中的命令。

在 Visual FoxPro 的操作中，窗口是重要的输入/输出界面。同时，Visual FoxPro 的对话框也有很多种，许多命令（在菜单中后面跟有...标记的命令）在执行时，要在屏幕上弹出一个对话框，以便提供一种“人机对话”的交互环境。凡是熟悉 Windows 环境操作的用户，就不会对 Visual FoxPro 的操作感到陌生。

#### 5.1.1 程序的建立与维护

程序文件又称命令文件，它是由 Visual FoxPro 的各条命令（或语句）按照一定的顺序和规则组织起来，以文件的形式存储在磁盘中，执行时再调入内存。

由于 Visual FoxPro 系统本身提供了文本编辑器，因此可以直接在集成环境中编写程序。可以利用菜单选择方式或在【命令】窗口中输入命令的方式来编写程序。

##### 1. 创建与修改程序文件

程序的建立与修改一般是通过 Visual FoxPro 系统提供的文本编辑器进行的。创建程序可以通过以下步骤进行。

- (1) 单击系统的【文件】菜单中的【新建】命令，屏幕将弹出【新建】对话框。在【新建】对话框中选定【程序】，然后单击【新建文件】按钮。
- (2) 打开文本编辑器的【程序 1】窗口，用户可以在这个窗口中输入程序，如图 5.1 所示。
- (3) 程序输入完后，可以从【文件】菜单中选择【保存/另存为】命令来保存文件，或按下 CTRL+W 组合键保存文件。

也可在 Visual FoxPro 的【命令】窗口中直接输入命令来建立和运行程序。在【命令】窗口中输入：MODIFY COMMAND <程序文件名>。随后可在打开的文本编辑器窗口中等待用户输入新内容。

2. 运行程序文件

程序存盘后，可以利用多种方式来多次执行它。为了提高运行速度，系统自动将.prg 文件生成一个.fxp 文件。程序运行结果显示在主窗口中，如图 5.2 所示。运行程序文件的步骤如下。

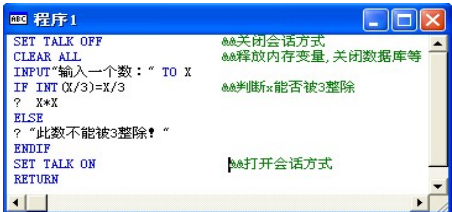


图 5.1 程序编辑窗口



图 5.2 程序及运行结果

- (1) 从【程序】菜单中选择【运行】命令，打开【运行】对话框。
- (2) 从【运行】对话框中选择一个要运行的程序，并单击【运行】按钮。也可以单击工具栏中的运行按钮。
- (3) 在【命令】窗口中键入：DO 程序文件名。

在执行一个 Visual FoxPro 程序时，根据需要可以在程序文件名前加上路径。例如：

```
DO d:\Visual FoxPro\程序1
```

该命令可以从【命令】窗口发出，也可以出现在某个程序文件中。这样就使得在执行一个程序的过程中还可以调用另一个程序。

当执行程序时，文件中包含的命令被依次执行，直到程序执行完毕，或遇到以下命令：

- CANCEL：终止程序运行，清除所有私有变量，返回【命令】窗口。
- DO：转去执行另一个程序。
- RETURN：结束当前程序的执行，返回到调用它的上级程序，若无上级程序则返回到【命令】窗口。
- QUIT：退出 Visual FoxPro 系统，返回到操作系统。

Visual FoxPro 程序文件通过编译、连编，可以产生不同的目标代码文件。如果没有对程序

进行编译，而且程序中存在错误，则在执行程序的过程中，Visual FoxPro 会指出程序中存在的错误，并出现错误提示对话框。此时，可以取消操作，终止程序的运行，修改后再运行。

### 3. 维护程序

修改程序文件时，可以从【文件】菜单中选择【打开】命令，从屏幕显示的【打开】对话框中选择要修改的文件，在文件编辑器窗口中进行编辑修改操作。

### 4. 编程时的一般规则

- ① 每行只能写一条命令，每条命令以回车换行作为结束标志。
- ② 如果一行写不下一条命令时，则可以分多行输入。在未输入完的语句后面加上分号，然后回车换行，系统会把此行作为前一行的续行。
- ③ “\*”、“&&”、“NOTE”用来作为程序注释的符号和命令；“\*”及“NOTE”只能放在一程序的前面，而“&&”可以放在前面或后面。

## 5.1.2 编程基础

一个程序一般都包括数据输入、数据处理、数据输出三部分。数据的输入和输出是编写程序时最常见的命令。所以在讨论程序控制结构之前，首先介绍几个常用的数据输入和输出命令。

### 1. 键盘输入语句

键盘输入命令有 ACCEPT、INPUT、WAIT 三条命令，这三条命令都是等待用户从键盘输入数据，并存入指定的内存变量中。

#### 1) ACCEPT命令

ACCEPT 命令的格式为

```
ACCEPT [<提示符表达式> TO <内存变量>]
```

其功能是等待用户在键盘上输入一个字符型数据，输入数据中不包含定界符，否则它将作为字符数据的一部分。

例如，将“李晓林”输入到 XM 变量中：

```
ACCEPT "请输入姓名：" TO XM
```

系统会提示“请输入姓名：”，用户可输入不带定界符的字符串，如下所示：

```
请输入姓名：李晓林
? XM
李晓林
```

上述带下划线的部分是上机输入内容，其他部分是执行后显示的内容。

#### 2) INPUT命令

INPUT 命令的格式为

```
INPUT [<提示符表达式>] TO <内存变量>
```

其功能是等待用户在键盘上输入表达式，并运算这个表达式，然后将运算结果存入指定变量中。当输入常量时，常量必须有定界符，否则可能出错。

例如，将“李晓林”输入到 `fn` 变量中：

```
INPUT "请输入姓名：" TO fn
```

系统会提示“请输入姓名：”，用户必须输入带定界符的字符串，如下所示：

```
请输入姓名： "李晓林"
? fn
李晓林
```

将逻辑值输入到 `TYL` 变量中以表示团员状况：

```
INPUT "团员否：" TO TYL
```

运行后即为

```
团员否： .T.
? TYL
.T.
```

### 3) WAIT 命令

WAIT 命令的格式如下：

```
WAIT [<提示符表达式>] [TO <内存变量>][WINDOWS[NOWAIT]][CLEAR];
[TIMEOUT<数值表达式>]
```

其功能是等待用户从键盘上输入一个字符并存入指定变量中。其中：

- `WINDOWS[NOWAIT]`：表示 WAIT 是一个单独的窗口，提供的信息将放在该窗口中。
- `CLEAR`：清除显示内容。
- `TIMEOUT<数值表达式>`：规定等待时间。

例如，将字符“W”输入到 `B` 变量中：

```
WAIT "请输入字符：" TO B      && 如果只有 WAIT，则显示"按任意键继续..."
```

运行后即为

```
请输入字符： W
? B
W
```

例如，在【命令】窗口输入下列命令：

```
WAIT "输入有误，请重新输入..." WINDOW TIMEOUT 5
```

执行该命令时，在主窗口右上角出现一个提示窗口，显示“输入有误，请重新输入...”，此时程序暂停执行。当用户按任意键或超过 5 秒钟，提示窗口自动关闭，程序继续执行。

可见，这个命令在没有指定存储数据的变量时，往往起到暂停程序运行的作用。一般用于程序的调试过程中。

## 2. 格式输出语句

无论是内存变量还是字段变量都可以使用“?”或“??”命令输出，这种输出方式简单，

但是不能按照用户指定的位置输出。格式输出语句能按用户的要求来设计屏幕格式，使之美观、方便。格式输出语句如下所示：

@ <行号, 列号> SAY <表达式>

其功能是从指定的行、列号开始输出表达式的值。其中：

- <表达式>可以是常量、字段变量及由它们组成的表达式。
- 定位输出时，一次只能输出一个表达式。

例如，在第 2 行、第 10 列输出“格式输出语句”：

@ 2, 10 SAY "格式输出语句"

### 3. 程序设计辅助语句

#### 1) CLEAR语句

格式：CLEAR

功能：清除屏幕信息。

#### 2) SET TALK ON/OFF语句

格式：SET TALK ON/OFF

功能：将执行的结果传送/不传送到屏幕上。

#### 3) 程序运行结束语句

|        |                                |
|--------|--------------------------------|
| QUIT   | && 返回操作系统                      |
| CANCEL | && 返回 Visual FoxPro 6.0 【命令】窗口 |
| RETURN | && 返回调用处                       |

### 4. 顺序结构

在结构化程序设计方法中，程序的基本控制结构有三种，即顺序结构、选择结构和循环结构。顺序结构是一种线形结构，是最基本的程序结构，它是按照命令或语句的排列顺序依次执行。下面给出一个简单的顺序结构的程序示例。

**例 5.1** 从键盘输入圆的半径  $r$ ，通过计算圆面积公式  $\pi r^2$  计算出圆的面积  $s$ ，最后输出该圆的面积  $s$ 。参考程序如下：

|                      |                     |
|----------------------|---------------------|
| * p1_1.prg           |                     |
| CLEAR                | && 清除主窗口原来显示的内容     |
| INPUT "圆的半径 r=" TO r | && 键盘输入一个数值，赋值给变量 r |
| s = 3.1416*r*r       | && 将计算结果存储到变量 s 中   |
| ? "圆的面积 s=", s       | && 显示字符串及变量 s 的值    |
| RETURN               | && 程序结束返回           |

运行程序：

DO p1\_1

运行结果显示：

圆的半径 r = 5  
圆的面积 s = 78.5400



## 练习与操作

**练习 1** 编写程序 PG1, 显示“学生表”中年龄大于 20 的记录并把专业编号是“03”、“04”的入学成绩增加 10 分。

```
SET TALK OFF
CLEAR
USE d:\学生成绩系统\DATA\学生表
d = YEAR(出生日期)
LIST ALL FOR d>=1988
REPL ALL 入学成绩 WITH 入学成绩+10 FOR "0" $ 专业编号
USE
SET TALK ON
RETURN
CTRL+W                && 存盘退出
```

运行程序:

```
DO PG1
```

**练习 2** 编写程序 PG2, 显示“学生表”中男生、女生的有关内容: 输入一个成绩数据, 查找表中入学成绩大于该数的一个记录, 把出生在 1989 年之前的所有学生的入学成绩加上 15 分。

```
SET TALK OFF
CLEAR
USE d:\liuli\abc\学生表
ACCE "请输入一个性别: " TO xb
LIST FOR 性别=XB
WAIT
INPUT "输入一个分数: " TO fs
LOCA FOR 入学成绩 > fs
DISP
WAIT
ACCE "输入一个日期: " TO csrq
REPL ALL 入学成绩 WITH 入学成绩+15 FOR YEAR(出生日期)<1989
CTRL+W    && 存盘退出
```

运行程序:

```
DO PG2
```

## 5.2 程序的流程控制

在结构化程序设计方法中, 程序基本控制结构有顺序结构、选择结构和循环结构。上面已经讲解了顺序结构的编程方式。下面主要介绍选择结构和循环结构。

### 5.2.1 选择结构

选择结构是根据给定的条件是否为真, 决定执行不同的分支, 并完成相应的操作。例如, 在求一元二次方程  $Ax^2 + Bx + C = 0$  的解时, 一般先求出  $D = B^2 + 4AC$ , 再根据  $D$  大于、等于或小于零, 分别求出方程的根为实根、重根或虚根。Visual FoxPro 系统提供了单条件选择结构和多分支条件选择结构。

## 1. 单条件选择结构

单条件选择结构的格式为

```
IF <条件表达式>
    <语句命令序列 1>
[ELSE <语句命令序列 2>]
ENDIF [注释]
```

如果<条件表达式>为真 (T.)，则执行<语句命令序列 1>，否则执行<语句命令序列 2>。如果没有[ELSE <语句命令序列 2>]选项，当<条件表达式>为假时，则执行 ENDIF 之后的语句。

**例 5.2** 从键盘上输入一个数，若该数能被 3 整除，则输出该数的平方。参考程序如下：

```
* p1_2.prg
SET TALK OFF                                && 关闭会话方式
CLEAR ALL                                  && 释放内存变量,关闭数据库等
INPUT "输入一个数:" TO x
IF INT(x/3)=x/3                             && 判断 x 能否被 3 整除
    ? x*x
ELSE
    ? "此数不能被 3 整除!"
ENDIF
SET TALK ON                                && 打开会话方式
RETURN
```

运行该程序时，输入 x 的值，根据 x 是否能被 3 整除决定是否显示 x 的平方值。

下面来看一个 IF...ELSE...ENDIF 结构的例子。

**例 5.3** 计算下列分段函数。

$$y = \begin{cases} x^2 + 10 & x < 5 \\ 10 - 10 & x \geq 5 \end{cases}$$

参考程序如下：

```
* p1_3.prg
SET TALK OFF
INPUT "x=" TO x
IF x<5
    y=x*x+10
ELSE
    y=10*x-10
ENDIF
? "y=",y
SET TALK ON
RETURN
```

另外，在选择结构中允许多层嵌套，即在 IF...ENDIF 选择结构中可以包含许多 IF...ENDIF 结构。

例 5.4 在“成绩表.dbf”文件中，查找并判断某学生的数学成绩是否及格，若及格则显示“及格”信息，若不及格则显示“不及格”信息。参考程序如下：

```
* p1_4.prg
CLEAR
CLEAR ALL
OPEN DATABASE data\学生           && 打开学生数据库文件
USE data\成绩表                   && 打开成绩表.dbf 文件
ACCEPT "输入学生学号:" TO xh      && 键盘输入数据，存储到 xh 中(均为字符型)
LOCATE ALL FOR 学号=xh             && 在表文件中顺序查找"学号=xh"的记录
IF NOT EOF( )                     && 判断记录指针是否移到结束标志
    IF 考试成绩>60                && 内层 IF
        ? "及格"
    ELSE
        ? "不及格"
    ENDIF                         && 内层 ENDIF
ELSE
    ? "学号输入错!"
ENDIF                             && 外层 ENDIF
USE                                && 关闭打开的表文件
RETURN
```

提示：在嵌套结构的程序设计中，要注意 IF 的出口语句 ENDIF 的个数及它所在的位置，IF 和 ENDIF 必须一一对应，外层的 IF...ENDIF 一定包含内层的 IF...ENDIF，不能交叉。另外，为使 IF 语句嵌套结构清晰，最好把程序写成层层缩进的形式。

2. 多分支条件选择结构

当要判断的条件很多，又要根据不同的条件执行不同的操作时，使用 IF...ENDIF 进行多重判断时虽能解决问题，但并不方便，且嵌套层数太多，容易出错。可以使用 DO CASE...ENDCASE 多分支条件选择结构来取代。DO CASE 命令结构如图 5.3 所示。多分支条件选择结构的格式如图 5.4 所示。

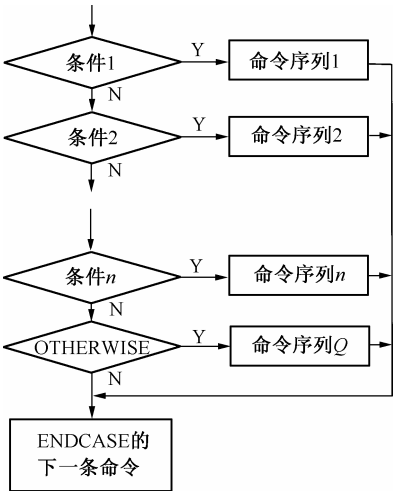


图 5.3 多分支条件选择结构



图 5.4 多分支条件选择格式

当系统执行 DO CASE 语句时，将逐个判断 CASE 后面的条件是否为真。只要遇到一个条件为真的 CASE 时，就执行其后的语句序列，语句序列执行完毕后，跳到 ENDCASE 后面的语句去执行。如果所有的 CASE 后面的条件都为假，则执行 OTHERWISE 后面的语句序列。如果没有 OTHERWISE 语句，则直接转去执行 ENDCASE 后面的语句。

**例5.5** 在“成绩表.dbf”文件中查找任意学生，根据其平均分判断该生的学习情况。分数段为 100~90、89~75、74~60、59~0，分别用优、良、及格和不及格来表示。参考程序如下：

```
* p1_5.prg
SET TALK OFF
CLEAR
CLEAR ALL
USE d:\学生成绩系统\data\成绩表.dbf
ACCEPT "学号: " TO xh
LOCATE ALL FOR 学号 = xh
IF .NOT. EOF( )
    DO CASE
        CASE 考试成绩<=100 AND 考试成绩>=90
            ? "成绩优秀"
        CASE 考试成绩>=75
            ? "成绩优良"
        CASE 考试成绩>=60
            ? "成绩及格"
        CASE 考试成绩>=0
            ? "成绩不及格"
        OTHERWISE
            ? "成绩计算错误"
    ENDCASE
ELSE
    ? "学号输入错"
ENDIF
USE
RETURN
```

## 练习与操作

**练习 1** 编写程序 PG3，某商品如果一次购买 100 件以上时，可享受 8% 的优惠。试编写程序根据输入的单价和数量来计算应付金额。参考程序如下：

```
SET TALK OFF
CLEAR
INPUT "数量: " TO s1
INPUT "单价: " TO dj
je = dj*s1
IF s1 >= 100
    je = je*0.92
ENDIF
```

```
? "应付金额:" + STR(je,6,2)
SET TALK ON
```

**练习 2** 编写程序 PG4, 判断公历某年是否为闰年。我们知道, 不能被 4 整除的年份肯定不是闰年、能被 4 和 100 整除但不能被 400 整除的也不是闰年, 其余皆是闰年。参考程序如下:

```
SET TALK OFF
CLEAR
INPUT "所查年份?" TO nf
IF MOD(nf,4)=0
    IF MOD(nf,100)=0
        IF MOD(nf,400)=0
            ? nf, '是闰年'
        ELSE
            ? nf, '不是闰年'
        ENDIF
    ELSE
        ? nf, '是闰年'
    ENDIF
ELSE
    ? nf, '不是闰年'
ENDIF
SET TALK ON
```

**练习 3** 编写程序 PG5, 计算整存整取应得利息。参考程序如下:

```
SET TALK OFF
CLEAR
INPUT "本金(元): " TO bj
INPUT "存期(年): " TO cq
DO CASE
    CASE cq>=8
        ll=8.4
    CASE cq>=5
        ll=7.5
    CASE cq>=3
        ll=6.9
    CASE cq>=2
        ll=6.6
    CASE cq>=1
        ll=6.3
    CASE cq>=0.5
        ll=4.5
    OTHERWISE
        ll=1.7
ENDCASE
?
? "应得利息: ", bj*cq*ll/1000
SET TALK ON
```

## 5.2.2 循环结构

循环结构是在给定的条件下反复执行某些相同的操作。被反复执行的操作称为循环体。Visual FoxPro 提供了三种循环语句：DO WHILE...ENDDO、FOR...ENDFOR 和 SCAN...ENDSCAN。

### 1. DO WHILE...ENDDO 循环

DO WHILE...ENDDO 循环的格式如下：

```
DO WHILE <条件表达式>
    <语句序列 1>
    [LOOP]
    <语句序列 2>
    [EXIT]
    <语句序列 3>
ENDDO [注释]
```

当 DO WHILE 语句中<条件表达式>的值为真时，反复执行 DO WHILE 与 ENDDO 之间的语句，直到<条件表达式>的值为假时，结束循环，执行 ENDDO 后面的语句。注意：

- LOOP 语句直接转到 DO WHILE 子句，而不执行 LOOP 和 ENDDO 之间的命令。LOOP 只能在循环结构中使用。
- EXIT 语句直接跳转到循环体之外，执行 ENDDO 后面的语句。EXIT 只能在循环结构中使用，即无条件结束循环语句。

例 5.6 求 1~100 之间的奇数之和，要求在程序中使用 LOOP 语句。参考程序如下：

```
* p1_6.prg
SET TALK OFF
CLEAR
s=1
n=1
DO WHILE n<100
    n=n+1
    IF int(n/2)=n/2    && 判断 n 是否整除
        LOOP
    ENDIF
    s=s+n              && 累加求和
ENDDO                 && 结束循环体
? "1~100 之间的奇数和是", s
SET TALK ON
```

为使程序最终跳出 DO WHILE 循环体，在程序循环过程中必须有修改循环条件的语句，否则程序将永远跳不出循环，这种情况称为无限循环。

## 2. FOR...ENDFOR 循环

FOR...ENDFOR 循环语句的格式如下：

```
FOR <控制变量> = <初值> TO <终值> [STEP<步长>]
    <语句序列>
    [LOOP]
    [EXIT]
ENDFOR [注释]
```

其功能是根据设定的循环次数，重复执行语句序列。执行 FOR 语句时，系统先把初值赋给循环控制变量。然后，判断循环控制变量是否大于（步长为正数）或小于（步长为负数）终值。若是，则结束循环，执行 ENDFOR 后面的语句；若不是，则执行循环体内的语句序列。执行到 ENDFOR 时，控制变量按步长递增或递减，根据循环控制变量和终值的大小，决定是否继续执行循环体。步长为 1 时可省略 STEP。

**例 5.7** 打印一个数据表中的全部字段名。其中字段数可用 FCOUNT() 函数得到，字段名可用 FIELDS() 函数得到。参考程序如下：

```
* p1_7.prg
CLEAR
USE d:\学生成绩系统\data\学生选课表.dbf
m=FCOUNT()
FOR n=1 TO m
    ? FIELDS(n)                && 显示各字段名
ENDFOR
```

**例 5.8** 使用 FOR...ENDFOR 循环求 1~100 之间的奇数之和。参考程序如下：

```
* p1_8.prg
s=0
FOR n=1 TO 100 STEP 2
    s=s+n
ENDFOR
? s
```

## 3. SCAN...ENDSCAN 循环

SCAN...ENDSCAN 循环语句的格式如下：

```
SCAN [FOR <条件表达式 1>[WHILE <条件表达式 2>]][<范围>]
    <语句序列>
    [LOOP]
    [EXIT]
ENDSCAN [注释]
```

其功能是对当前数据表文件中满足条件的记录进行加工处理。这是 Visual FoxPro 系统专门用于数据处理的循环命令，它自动移动记录指针，当程序执行到 ENDSCAN 或 LOOP 语句时，会对<条件表达式 1>或<条件表达式 2>进行判断，如果条件成立，它会自动将指针移到下一个符合条件的记录上。

**例 5.9** 显示“学生选课表.dbf”文件中课程编号为“CC02”、成绩在“70”以上的记录。参考程序如下：

```
* p1_9.prg
CLEAR
USE d:\学生成绩系统\data\学生选课表.dbf
SCAN FOR 课程编号="CC02" AND 成绩>70      && 扫描满足条件的记录
    DISPLAY
    WAIT                                     && 显示当前记录
ENDSCAN
USE
```

执行该程序时，把“成绩表.dbf”文件中所有满足条件的记录显示在屏幕上。

以上介绍的三种循环结构的例子都是单层循环，有时根据解决问题的需要，要用到两层或多层循环结构，即在一个循环中又包含另一个循环，这种结构称为循环嵌套。

**提示：**① 循环开始语句和结束语句必须成对出现。

② 循环结构只能嵌套，不能交叉，即内层循环必须包含在外循环之内。

③ 各层次的循环控制变量不要重名，以免混淆。

## 练习与操作

**练习 1** 编写程序 PG6，使用 DO WHILE 语句计算 1~100 之间的奇数的平方和。参考程序如下：

```
SET TALK OFF
CLEAR
s=0
x=1
DO WHILE x<100
    s=s+x^2
    x=x+2
ENDDO
? s
SET TALK ON
```

**练习 2** 编写程序 PG7，显示“学生表”中的记录，显示完一个暂停一下。参考程序如下：

```
SET TALK OFF
USE data\学生表
DO WHILE NOT EOF()
    DISP
    WAIT
    SKIP
ENDDO
CLOSE ALL
SET TALK ON
```

**练习 3** 编写程序 PG8，在“学生表”中找出入学成绩大于某指定值的记录并显示出来。参考程序如下：

```
SET TALK OFF
```



```
USE d:\学生成绩系统\data\学生表
INPUT "请输入一个数: " TO aa
DO WHILE NOT EOF()
    IF 入学成绩<=aa
        SKIP
        LOOP
    ENDIF
    DISP
    SKIP
ENDDO
USE
SET TALK ON
```

**练习 4** 编写程序 PG9，在“学生表”中显示第一个入学成绩大于某数的记录。参考程序如下：

```
SET TALK OFF
USE d:\学生成绩系统\DATA\学生表
INPUT "输入一个数: " TO bb
DO WHILE NOT EOF()
    IF 入学成绩>bb
        DISP
        EXIT
    ENDIF
    SKIP
ENDDO
CLOSE ALL
SET TALK ON
```

**练习 5** 编写程序 PG10，打印“学生表”中的全部字段名。其中字段数可用 FCOUNT( )函数得到，字段名可用 FIELDS()函数得到。参考程序如下：

```
SET TALK OFF
CLEAR
USE DATA\学生表
m=FCOUNT()
FOR n=1 TO m
    ? FIELDS(n)                && 显示各字段名
ENDFOR
USE
SET TALK ON
```

**练习 6** 编写程序 PG11，计算九九乘法表，运行结果如图 5.5 所示。参考程序如下：

```
SET TALK OFF
CLEAR
i=1
DO WHILE i<=9
    j=1
```

```

DO WHILE j<=9
    IF j<=i
        ?? STR(i,1),'*',STR(j,1),'=',STR(i*j,2),' '
    ENDIF
    j=j+1
ENDDO
?
i=i+1
ENDDO
SET TALK ON

1 * 1 = 1
2 * 1 = 2 2 * 2 = 4
3 * 1 = 3 3 * 2 = 6 3 * 3 = 9
4 * 1 = 4 4 * 2 = 8 4 * 3 = 12 4 * 4 = 16
5 * 1 = 5 5 * 2 = 10 5 * 3 = 15 5 * 4 = 20 5 * 5 = 25
6 * 1 = 6 6 * 2 = 12 6 * 3 = 18 6 * 4 = 24 6 * 5 = 30 6 * 6 = 36
7 * 1 = 7 7 * 2 = 14 7 * 3 = 21 7 * 4 = 28 7 * 5 = 35 7 * 6 = 42 7 * 7 = 49
8 * 1 = 8 8 * 2 = 16 8 * 3 = 24 8 * 4 = 32 8 * 5 = 40 8 * 6 = 48 8 * 7 = 56 8 * 8 = 64
9 * 1 = 9 9 * 2 = 18 9 * 3 = 27 9 * 4 = 36 9 * 5 = 45 9 * 6 = 54 9 * 7 = 63 9 * 8 = 72 9 * 9 = 81

```

图 5.5 九九乘法表

**练习 7** 编写程序 PG12, 请用 SCAN...ENDSCAN 循环编写判断学生成绩“优、良、中、差”的程序。参考程序如下:

```

SET TALK OFF
USE d:\学生成绩系统\data\成绩表
STORE 0 TO n1,n2,n3,n4
SCAN
    cj=平时成绩*0.2+考试成绩*0.8
    DO CASE
        CASE cj>=90
            REPLACE 等级 WITH "优"
            n1=n1+1
        CASE cj>=75
            REPLACE 等级 WITH "良"
            n2=n2+1
        CASE cj>=60
            REPLACE 等级 WITH "中"
            n3=n3+1
        CASE cj<60
            REPLACE 等级 WITH "差"
            n4=n4+1
    ENDCASE
ENDSCAN
? "优,良,中,差为: ",n1,n2,n3,n4
USE
SET TALK ON
RETURN

```

## 5.3 模块结构程序设计

### 5.3.1 过程及过程调用

Visual FoxPro 与其他高级语言一样，支持结构化程序设计方法，允许将若干命令或语句组合在一起作为整体调用，这些可独立存在并可整体调用的命令语句组合称为过程。

#### 1. 外部过程

外部过程也称为子程序，和主程序一样是以程序文件（.prg）的形式单独存储在磁盘上。子程序是指能被其他程序调用的程序。通常情况下，被调用的称为子程序，调用它的称为主程序。子程序中必须使用的一条命令是返回命令 RETURN，格式为

```
RETURN [<表达式>|TO MASTER|TO <程序文件名>]
```

其功能是返回到调用该子程序的上级程序。例如，建立如下的程序文件：

```
*正在执行主程序 p1_10.prg
SET TALK OFF
CLEAR
DO 程序\pg11
SET TALK ON
*pg11.prg
? "正在执行 pg11"
RETURN
```

#### 2. 内部过程

将多个过程组织在一个文件中（这个文件称为过程文件），或者把过程放在调用它的程序文件的末尾。

Visual FoxPro 为了识别过程文件或程序文件中的不同过程，规定过程文件或程序文件中的过程必须用 PROCEDURE 声明，从而建立过程文件。

#### 3. 过程文件的建立

过程文件的创建格式如下：

```
*过程文件名 1
PROCEDURE <过程名 1>
<命令序列 1>
RETURN
...
PROCEDURE <过程名 n>
<命令序列 n>
RETURN
```

过程文件中的每个子过程必须以 PROCEDURE 开头，后面跟过程名，每个子过程以 RETURN 结束。

调用过程文件之前，应先打开相应的过程文件，格式如下：

在调用过程前，一旦打开过程文件，那么其中的所有子过程都被打开，可以随时调用。Visual FoxPro 中，如果要同时打开两个以上的过程文件，可以在过程文件名之间用逗号分隔。如果分别打开多个过程文件，则后打开的过程文件将会关闭先前所打开的过程文件，为避免这种情况发生，在 SET PROCEDURE 命令中加入 ADDITIVE 参数。

调用过程和调用子程序一样，使用“DO <过程名>”命令来调用指定的子过程。

调用过程文件的格式如下:

其功能是执行以<过程名>为名的过程，即执行已有的过程。

返回过程文件的格式如下:

其功能是将控制返回到调用程序中调用命令的下一条语句，即返回过程的调用处。

当过程文件调用结束后，应及时关闭过程文件。可以使用下列命令关闭过程文件：

- SET PROCEDURE TO
- CLOSE PROCEDURE

**例 5.10** 利用过程文件分别计算圆的面积和球的体积。主程序如下:

[illegible]

```
s=4/3*3.1416*r^3
? "球体积=",s
RETURN
```

在磁盘上分别建立以上两个文件，一个是“主程序.prg”。当运行“主程序.prg”后，则打开过程文件“p1\_11.prg”，分别调用相应的子过程来求圆面积和球体积。

**例 5.11** 建立如下程序文件，将过程放在程序文件中。

```
* p1_12.prg
SET TALK OFF
? "正在执行主程序"
DO d:\学生成绩系统\程序\pg11
SET TALK ON
PROCEDURE pg11
? "正在执行 pg11"
RETURN
```

**例5.12** 利用过程文件实现对“学生成绩系统”数据库的“学生表”进行查询、删除和插入操作。主程序如下：

```
* p1_13.prg                                && 主程序文件名
SET TALK OFF
CLEAR
OPEN DATABASE data\学生
SET PROCEDURE TO d:\学生成绩系统\程序\pg7    && 打开过程文件
USE data\学生表.dbf EXCLUSIVE                 && 表文件以独占方式打开
INDEX ON 姓名 TO xm
DO WHILE .T.                                  && 显示菜单
    CLEAR
    @ 2,20 say "学生管理系统"
    @ 4,20 say "a:按姓名查询"
    @ 6,20 say "b:按记录号删除"
    @ 8,20 say "c:插入新的记录"
    @ 10,20 say "d:退出"
    ch=" "
    @ 12,20 say "请选择 a、b、c、d: " get ch
    READ
    DO CASE
        CASE ch="a"
            DO PROCE1
        CASE ch="b"
            DO PROCE2
        CASE ch="c"
            DO PROCE3
        CASE ch="d"
            EXIT
    ENDCASE
ENDDO
```

```

SET PROCEDURE TO                                && 关闭过程文件
CLOSE DATABASE
SET TALK ON

```

过程文件如下:

```

*proce.prg                                     && 过程文件名
PROCEDURE PROCE1                               && 查询过程
CLEAR
ACCEPT  "请输入姓名: " TO name
SEEK name
IF FOUND( )
    DISPLAY
ELSE
    ? "查无此人!"
ENDIF
WAIT
RETURN
PROCEDURE proce2                               && 删除记录过程
CLEAR
INPUT  "请输入要删除的记录号:" TO n
GO n
DELETE
WAIT "物理删除吗 y/n:" TO flag
IF flag="y" .OR. "y"
    PACK
ENDIF
RETURN
PROCEDURE proce3                               && 插入新的记录过程
CLEAR
APPEND
RETURN

```

### 5.3.2 过程调用中的参数传递

过程可以没有或有多多个参数，多个参数由逗号隔开。下面讨论形式参数定义和参数传递。

#### 1. 有参过程中的形式参数定义

有参过程中的形式参数定义语句为

```
PARAMETERS <参数表>
```

该语句必须是过程中的第一条语句。<参数表>中的参数可以是任意合法的内存变量名。

#### 2. 程序与被调用过程间的参数传递

程序与被调用过程间的参数传递是通过过程调用语句“DO <过程名> WITH <参数表>”中的“WITH <参数表>”子句实现的，具体格式如下：

DO<文件名>|<过程名> WITH <参数表>

其中, DO 命令的<参数表>中的参数称为实际参数(实参), PARAMATERS 命令的<参数表>中的参数称为形式参数(形参)。两个<参数表>中的参数必须相容,即个数相同,类型和位置一一对应。实际参数可以是任意合法表达式,形式参数是过程中的局部变量,用来接收对应实际参数的值。

参数的传递方式有以下两种。

① 按值传递。按值传递时,传递给被调用过程(子过程)的参数是调用过程(父过程)调用时所使用的值。一般实参是一个变量或是一个表达式。

② 按地址传递。如果实参是一个变量(而不是表达式),则传递给形参的是该变量的地址。这时形参和实参是同一个变量,在过程中改变形参的值,会同时改变实参的值。默认模式下,变量利用按地址传递方式向过程传递,并利用按值传递方式向函数传递。

使用上述命令时,需要注意以下几点。

① 该命令必须放在本级程序的首行。

② 必须和 DO...WITH 配合使用。<内存变量表>中变量的个数要与上级程序中的 WITH <参数表>中的参数个数相同。各变量用逗号分隔,最多能传递 24 个参数。

③ 变量类型自动与上级程序中的参数相匹配。

④ 参数传递有两种方式:按值传递和按地址传递。如果使用按值传递方式,则子程序中参数变化后的值不回传给上级调用程序。如果使用按地址传递方式,则子程序中参数变化后的值要回传给上级调用程序。常量和表达式只能使用按值传递方式,内存变量既可以使用按值传递方式,又可以使用按地址传递方式。使用按值传递方式的变量要用括号括起来,使用按地址传递方式的变量不加括号。如果不允许子程序改变传递参数变量的值,应该使用按值传递方式。如果允许子程序改变传递参数变量的值,则要使用按地址传递方式。

**例 5.13** 给出下列程序的输出结果。

```
* p1_14.prg
SET TALK OFF

x=1
y=3
DO SUB WITH x,(y),5
? x,y
RETURN

PROCEDURE sub
PARAMETER a,b,c
a=a+b+c
b=a+b-c
RETURN
```

输出结果是

例 5.14 利用参数传递操作进行编程，计算圆的面积。

```
* p1_15.prg
SET TALK OFF
CLEAR
s=0
INPUT "请输入圆的半径:" TO r
DO area WITH r,s
? "圆的面积为:",s
SET TALK ON
?
PROCEDURE area                && 计算面积的过程
PARA x,y                      && 形参说明
y=3.1416*x**2
RETURN
SET TALK ON
```

### 5.3.3 变量的作用域

程序设计离不开变量，如果以变量的作用域进行划分，可分为公共变量、私有变量和局部变量。

#### 1. 公共变量

公共变量是指在所有程序模块中都可以使用的内存变量。公共变量要先建立后使用。定义公共变量的格式如下：

```
PUBLIC <内存变量表>
```

该命令的功能是建立公有的内存变量，并为它们赋初值：逻辑假“F.”。

对于公共变量，需要注意以下几点。

- ① 当定义多个变量时，各变量名之间用逗号隔开。用 PUBLIC 命令定义的内存变量，在程序执行期间可以在任何层次的程序模块中使用。
- ② 变量定义语句要放在使用此变量的语句之前，否则会出错。
- ③ 任何已经定义为公共变量的变量，可以用 PUBLIC 命令再定义，但不允许重新定义为局部变量。
- ④ 使用公共变量可以增强模块间的通用性，但会降低模块间的独立性。

#### 2. 局部变量

局部变量是指在建立它的程序及此程序调用的程序中有效的内存变量。定义局部变量的格式如下：

```
LOCAL <内存变量表>
```

该命令的功能是建立指定的内存变量，并为它们赋初值：逻辑假“F.”。



对于局部变量，需要注意以下几点。

① 由于 LOCAL 命令与 LOCATE 命令的前 4 个字母相同，所以这条命令的动词不能缩写。

② 在程序中没有声明为公共变量的内存变量都被看成是局部变量。

在子程序中可以用 PRIVATE 命令隐藏主程序中可能存在的变量，使这些变量在子程序中暂时无效。命令格式如下：

```
PRIVATE <内存变量表>
```

或者：

```
PRIVATE ALL [LIKE | EXCEPT <通配符>]
```

使用 PRIVATE 命令时，需要注意以下几点。

① 使用 PRIVATE 命令声明的内存变量，只能在本程序及其调用过程中使用。退出程序时，自动释放变量。

② 使用 PRIVATE 命令在过程中声明的局部变量，可以与上层调用程序出现的内存变量同名。但它们是不同的变量，在被调用的期间，上层过程中的同名变量将被隐藏。

### 3. 私有变量

在程序中直接使用而由系统自动建立的变量都是私有变量。私有变量的作用域是建立它的模块及其下属的各层模块。一旦建立它的模块程序运行结束，这些私有变量将自动清除。定义私有变量的格式如下：

```
PRIVATE [<内存变量表>] [ALL[LIKE|EXCEPT<通配符>]]
```

## 5.3.4 自定义函数

Visual FoxPro 为用户提供了几百个内部标准函数，但是并不能完全符合每个用户的需要，因此，为了程序设计的需要，必须自行设计函数，用户按一定规则来定义的函数，这种函数称为自定义函数。自定义函数的格式如下：

```
FUNCTION <函数名称>(<变量名称>)  
<语句序列>  
RETURN [<返回值>]  
ENDFUNC
```

定义了函数之后，可将它保存在单独的程序文件中，也可放在一般程序的底部，但不能将可执行的主程序放在函数之后。调用自定义函数与一般内部函数相同，函数执行后给调用程序返回一个数据，<返回值>可以是常量、变量或表达式等。如果在 RETURN 命令后没有加入返回值，Visual FoxPro 将自动返回.T。当程序或用户自定义函数执行到 RETURN 命令时就会立刻返回到调用程序中。

### 例 5.15 利用自定义函数，求 $x! + y!$ 。

```
* p1_16.prg
CLEAR
INPUT "输入 x 的值: " TO x
u=jc(x)                                && 调用函数 jc
INPUT "输入 y 的值: " TO y
v=jc(y)
?"sum=",u+v
FUNCTION jc
PARAMETERS n
s=1
IF n>=1
    FOR i=1 TO n
        s=s*i
    NEXT
ENDIF
RETURN (s)
ENDFUNC
```

在该例中定义了一个求阶乘的函数。函数名为 jc， $x!$  的求法是在主程序内从键盘输入 x 的值，再调用 jc 函数求得  $x!$  后将值带回主程序。最后将两个阶乘值相加。

可以使用以下几种方法来调用函数：

- ?DEMOF(7)                                && 打印函数的返回值
- a=DEMOF( )                               && 使用变量接收函数的返回值
- =DEMOF( )                               && 执行函数，不接收返回值

函数的返回值也可以有以下几种方式：

- RETURN abcd                             && 变量
- RETURN xy/0.15                         && 表达式
- RETURN .F.                               && 常量
- RETURN myfunc1( )                      && 自定义函数
- RETURN day(date( ))                   && 内部函数

在程序设计时，应该使用子程序还是函数？一般的做法是当程序代码比较多、工作重复且不需要返回值时，使用子程序；而函数最重要的是其返回值。因此有没有返回值，经常是决定使用子程序或函数的关键。

## 练习与操作

**练习 1** 利用主、子程序调用的方法，编写一个求 100 之内所有素数的程序 PG13.prg。参考步骤如下：

- (1) 在【命令】窗口中执行“MODIFY COMMAND PG13”命令。
- (2) 在打开的程序编辑窗口中，输入主程序的源代码如下：

```
*主程序 PG13.prg
SET TALK OFF
FOR m=3 TO 100 STEP 2
    n=int(SQRT(m))
    DO d:\学生成绩系统\程序\sub1.prg
ENDFOR
SET TALK ON
```

(3) 执行【文件】→【保存】命令，将 PG13.prg 程序存盘，然后关闭该编辑窗口。

(4) 在 Visual FoxPro 【命令】窗口中执行“MODIFY COMMAND sub1”命令。

(5) 在打开的程序编辑窗口中，输入子程序的源代码如下：

```
*子程序 sub1.prg
FOR i=3 TO n STEP 2
    IF MOD(m,1)=0
        RETURN
    ENDIF
ENDFOR
?? m
RETURN
```

(6) 执行【文件】→【保存】命令，将 sub1.prg 程序存盘，然后关闭该编辑窗口。

(7) 在【命令】窗口中输入“DO PG13”命令执行主程序。

**练习 2** 利用主、子程序调用的方法，编写一个可以求解  $s = (x*y)! - x! - y! - z!$  的程序 PG14.prg。参考步骤如下：

(1) 在【命令】窗口中执行“MODIFY COMMAND PG14”命令。

(2) 在打开的程序编辑窗口中，输入主程序的源代码如下：

```
*主程序 PG14.prg
SET TALK OFF
CLEAR
INPUT "x=" TO x
INPUT "y=" TO y
z=x*y
DO d:\学生成绩系统\程序\sub2.prg WITH x
DO d:\学生成绩系统\程序\sub2.prg WITH y
DO d:\学生成绩系统\程序\sub2.prg WITH z
? "s="+str(z-x-y,8)
SET TALK ON
RETURN
```

(3) 执行【文件】→【保存】命令，将 PG14.prg 程序存盘，然后关闭该编辑窗口。

(4) 在【命令】窗口中执行“MODIFY COMMAND sub2”命令。

(5) 在打开的程序编辑窗口中，设置子程序的源代码如下：

```
*子程序 sub2.prg 求 n!
PARAMETERS n
i=n-1
```

```
DO WHILE i>0
    n=n*i
    i=i-1
ENDDO
RETURN
```

(6) 执行【文件】→【保存】命令，将 sub2.prg 程序存盘，然后关闭该编辑窗口。

(7) 在【命令】窗口中输入“DO PG14”命令执行主程序。

**练习 3** 编写一个程序 PG15.prg，可以实现在读入一串数据后，自动将其逆向输出。参考步骤如下：

(1) 在【命令】窗口中执行“MODIFY COMMAND PG15”命令。

(2) 在打开的程序编辑窗口中，输入主程序的源代码如下：

```
*PG15.prg
SET TALK OFF
CLEAR
INPUT "请输入数据的个数：" TO n
DIMENSION x(n)
FOR i=1 TO n
    INPUT "请输入数据：" +str(0,3) TO x(i)
ENDFOR
@ ROW( )+2,1 SAY "输出的数据为："
FOR i=n TO 1 STEP -1
    ?? LTRIM(STR(x(i)))+""
ENDFOR
SET TALK ON
```

(3) 执行【文件】→【保存】命令，将 PG15.prg 程序存盘，然后关闭该编辑窗口。

**练习 4** 编写一个自定义函数程序 PG16.prg，利用自定义函数，求  $x!+y!$ 。参考步骤如下：

(1) 在【命令】窗口中执行“MODIFY COMMAND PG16”命令。

(2) 在打开的程序编辑窗口中，输入主程序的源代码如下：

```
*PG16.prg
SET TALK OFF
CLEAR
INPUT "输入 x 的值：" TO x
u=jc(x)                                && 调用函数 jc
INPUT "输入 y 的值：" TO y
v=jc(y)
?"sum=",u+v
FUNCTION jc
    PARAMETERS n
    s=1
    IF n>=1
        FOR i=1 TO n
            s=s*i
        NEXT
```

```
ENDIF  
RETURN (s)  
ENDFUNC  
SET TALK ON
```

(3) 将以上函数程序保存为 PG16.prg 文件后, 即可被其他程序调用或在【命令】窗口中直接调用。

## 思考与练习 5

### 一、思考题

1. 在 Visual FoxPro 6.0 中, 循环语句有哪几种? 分别是什么?
2. 何为外部过程? 何为内部过程?
3. 结构化程序设计的结构有哪几种? 各自的特点是什么?
4. 何为参数传递?
5. 按变量的作用域可将变量分为哪几类?

### 二、选择题

1. 执行 WAIT "选择" TO ANS 命令后, 内存变量 ANS 的类型是\_\_\_\_。  
A. 字符型                      B. 数值型                      C. 日期型                      D. 逻辑型
2. Visual FoxPro 的应用程序由三种基本结构组合而成, 它们是\_\_\_\_。  
A. 顺序结构、选择结构和循环结构                      B. 顺序结构、循环结构和模块结构  
C. 逻辑结构、物理结构和程序结构                      D. 分支结构、重复结构和子程序结构
3. 在非嵌套程序结构中, 可以使用 LOOP 和 EXIT 语句的基本程序结构是\_\_\_\_。  
A. TEXT...ENDTEXT                      B. DO WHILE...ENDDO  
C. IF...ENDIF                      D. DO CASE...ENDCASE
4. 有如下的 Visual FoxPro 程序:

```
SET TALK OFF  
m=0  
n=100  
DO WHILE n>m  
    m=m+n  
    n=n-10  
ENDDO  
? m  
RETURN
```

运行此程序显示 m 的值是\_\_\_\_。

- A. 0                      B. 10                      C. 100                      D. 99
5. 下面关于过程调用的陈述中, \_\_\_\_是正确的。  
A. 实参与形参的数量必须相等  
B. 当实参的数量多于形参的数量时, 多余的实参被忽略  
C. 当形参的数量多于实参的数量时, 多余的形参取逻辑假  
D. 上面 B 和 C 都对

6. 如果一个过程不包含 RETURN 语句, 或者 RETURN 语句中没有指定表达式, 那么该过程\_\_\_\_\_。
- A. 没有返回值      B. 返回 0      C. 返回 .T.      D. 返回 .F.
7. 在程序中定义局部变量的命令动词是\_\_\_\_\_。
- A. PUBLIC      B. PRIVATE      C. LOCAL      D. DECLARE
8. 有如下程序:

```
INPUT "a=" TO a
IF a=10
    s=0
ENDIF
s=1
? s
```

假定从键盘输入的 a 的值一定是数值型, 那么上面条件选择程序的执行结果是\_\_\_\_\_。

- A. 0      B. 1      C. 由 a 的值决定      D. 程序出错
9. 执行如下程序:

```
SET TALK OFF
s=0
i=1
INPUT "n=" TO n
DO WHILE s<=n
    s=s+i
    i=i+1
ENDDO
? s
SET TALK ON
```

如果输入值为 5, 则最后 S 的显示值是\_\_\_\_\_。

- A. 1      B. 3      C. 5      D. 6
10. 有程序如下:

```
USE cj
m.zf=0
SCAN
    m.zf=m.zf+zf
ENDSCAN
? m.zf
RETURN
```

其中数据库文件 cj.dbf 中有 2 条记录, 内容如下:

|   | xm | zf     |
|---|----|--------|
| 1 | 李四 | 500.00 |
| 2 | 张三 | 600.00 |

运行该程序的结果应当是\_\_\_\_\_。

- A. 1100.00      B. 1000.00      C. 1600.00      D. 1200.00

### 三、填空题

1. 定义公共变量使用命令\_\_\_\_，定义私有变量使用命令\_\_\_\_，定义局部变量使用命令\_\_\_\_。

2. n=6

```
s=1
DO WHILE n>1
    n=n-1
    s=s*n
ENDDO
? "s=",s
```

运行结果是\_\_\_\_\_。

3. SET TALK OFF

```
CLEAR ALL
x=9
IF int(x/3)=x/3
    ? x*x
ENDIF
SET TALK ON
RETURN
```

运行结果是\_\_\_\_\_。

4. SET TALK OFF

```
CLEAR
FOR k=1 TO 4
    FOR i=1 TO 2*k-1
        ?? "*"
    NEXT
    ?
NEXT
SET TALK ON
RETURN
```

运行结果是\_\_\_\_\_。

5. LOCAL nvar

```
nvar=myfunc1(16)
WAIT WINDOWS "16 平方根为："+STR(m.nvar)
FUNCTION myfunc1
    * 该程序计算数据的平方根
    PARA x
    RETURN sqrt(x)
```

运行结果是\_\_\_\_\_。

6. 执行 FOR...ENDFOR 语句时，若步长为\_\_\_\_\_值，则循环条件为“(循环变量)≤(终值)”；若步长为\_\_\_\_\_值，则循环条件为“(循环变量)≥(终值)”。

7. 在 Visual FoxPro 中, 结构化程序设计的三种基本逻辑结构是\_\_\_\_\_。
8. 在 Visual FoxPro 中, \_\_\_\_\_语句实现一种扩展的选择结构, 它可以根据条件从多组代码中选择一组执行。
9. 假设主过程和子程序中都建立了同一个变量, 为了避免子程序的运行在无意间改变主程序中的变量的取值, 可以用\_\_\_\_\_使此变量在子程序中暂时无效。
10. 在 Visual FoxPro 中, 程序文件的扩展名为\_\_\_\_\_。



## 第 6 章 表 单

表单是 Visual FoxPro 为用户提供的—个功能强大、操作方便的界面设计工具。利用表单设计器可以方便、快捷地设计出美观、友好的界面，同时也为用户进行数据操作提供了方便。表单是 Visual FoxPro 中最为常见的数据显示及编辑界面。

### 6.1 面向对象程序设计简介

Visual FoxPro 支持面向对象程序设计 (Object-Oriented Programming, OOP)。面向对象的程序设计方法与编程技术不同于面向过程程序设计，用户在程序设计时，主要考虑如何创建对象，并利用对象来简化程序设计。

#### 6.1.1 面向对象程序设计中的基本概念

面向对象程序设计是目前程序设计方法的主流，也是程序设计在思维和方法上的一次巨大进步。面向对象程序设计实际上是在“组装”程序，每个对象中的代码对于许多编程人员来说是透明的，编程人员更为关心的是功能和接口，也就是对象所具有的属性和方法。

##### 1. 对象 (object)

生活中对象是指各种大大小小的具体客观事物。但在程序设计中，对象是私有数据和对这些数据进行处理的操作 (方法) 相结合的程序单元 (实体)。可以将一个表单看成一个对象，也可以将表单中的一个命令按钮或—张图片看成对象。

##### 2. 属性 (property)

属性定义了对象所具有的数据，它是对象所有特征数据的集合。每个对象都具有“属性”。“属性”值可以在设计阶段设置，也可以在运行阶段更改。但有些“属性”是只读属性，不可改变。

##### 3. 方法 (method)

方法是指对象为实现—定功能而编写的代码。

##### 4. 事件 (event)

事件是由用户或系统的动作所引发的情况，由用户或者系统的操作而激活。Visual FoxPro 中的事件通常包括键盘“事件”和鼠标“事件”，例如单击鼠标就发生了—个“Click”事件。为了响应事件，可以为事件加入相应的代码，也可以执行某个方法。

## 5. 类 (class)

类是一组具有相同特性的对象的抽象定义。类是具有相同或相似特征的对象抽象，对象是类的具体的实例。类可以具有子类。

子类可以继承父类所有的属性和方法，也可以根据需要加入新的属性和方法。

### 6.1.2 Visual FoxPro中的类

#### 1. 类的概念

##### 1) 类与对象

在面向对象程序设计中，类与对象都是应用程序的组成模块。类是已经定义了的关于对象的特征和行为的模板。在表单控件工具栏中，每个控件按钮都代表一个类，使用其中某个按钮在表单上创建的一个控件就是一个对象。可见，类是对象的定义。类规定并提供了对象具有的属性、事件和方法。对象通过类来产生，对象是类的实例。

##### 2) 基类 (base class)

基类是 Visual FoxPro 预先定义的类，在【新建类】对话框的【派生于】下拉列表中，包含了全部基类，例如表单 (Form)、表单集 (FormSet) 等。基类可作为用户定义类的基础。用户可从基类来创建新类，并增加自己需要的新功能。

##### 3) 子类 (subclass)

以某个类的定义为起点创建的新类称为子类，前者称为父类。例如从基类创建新类时，基类是父类，新类是子类。

新类将继承父类的全部特征，包括对父类所做的任何修改。

#### 2. 类的特征

类是模板，它规定了各类对象的属性、事件和方法。此外，类还具有类的封装性、继承性、多态性、抽象性四大特征。

##### 1) 封装性

封装指包含并隐藏对象信息，如内部数据结构、对象的方法和属性代码。封装隐藏了对象内部的细节，例如对一个命令按钮设置 Caption 属性时，不必了解标题字符串是如何存储的。隐藏对象信息的优点有

- 有利于对复杂对象进行的管理。由于隐藏了对象内部细节，使用户能集中精力来使用对象的特性。
- 有利于程序的安全性。隐藏对象信息能防止代码不慎受到破坏。

##### 2) 继承性

继承性包括以下内容：

- 对象能自动继承创建它的类的功能。
- 子类能自动继承父类的功能。
- 对一个类的改动能自动反映到它的所有子类中。

继承性不但节省了用户的时间和精力，同时也减少了维护代码的难度。所以，继承性是合理地进行代码维护的重要措施。

### 3) 多态性

指一些关联的类包含同名的方法，但方法的内容可以不同，具体调用在运行时根据对象的类确定。

### 4) 抽象性

指提取一个类或对象与众不同的特征，而不对该类的所有信息进行封装处理。

### 5) 子类特征

一个子类可以拥有其父类的全部功能，但也可以增加自己的属性和方法，使它具有与父类不同的特殊性。

如果创建一个合适的子类，并在多处创建它的实例，就能使代码得到重复使用，因此定义子类是减少代码的途径之一。

## 3. Visual FoxPro的基类

Visual FoxPro 中的基类又可以分为容器类和控件类。

### 1) 容器类

可以包含其他对象的类称为容器类。容器对象可作为父对象，其包含的对象称为子对象。例如，表单对象作为容器，可以包含命令按钮、文本框、复选框等子对象。容器内还可以包含容器类对象，例如表单容器内包含表格、页框、命令按钮组等容器对象。而子容器中还可以包含命令按钮、选择按钮等控件对象。表 6.1 列出了基类中主要的容器类。

表 6.1 Visual FoxPro 中的容器类

| 容器类名称                        | 说明                            |
|------------------------------|-------------------------------|
| 列 (Column)                   | 可以容纳标头等对象，但不能容纳表单、表单集、工具栏和计时器 |
| 命令按钮组 (Command Button Group) | 只能容纳命令按钮                      |
| 表单 (Form)                    | 可以容纳页框、容器控件、容器或自定义对象          |
| 表单集 (FormSet)                | 可以容纳表单、工具栏                    |
| 表格 (Grid)                    | 只能容纳表格                        |
| 选择按钮组 (Option Button Group)  | 只能容纳选项按钮                      |
| 页面 (Page)                    | 只能容纳控件、容器和自定义对象               |
| 页框 (PageFrame)               | 只能容纳页框                        |
| 工具栏 (ToolBar)                | 可容纳任意控件、页框和容器                 |

### 2) 控件类

不允许包含其他对象的类称为控件类，换句话说，控件对象不能作为父对象。例如，命令按钮、选择按钮、复选框、文本框、标签等控件对象，就不能包含其他对象。表 6.2 列出了 Visual FoxPro 中常用的控件类。

表 6.2 Visual FoxPro 中的控件类

| 控件类名称                 | 说明                    |
|-----------------------|-----------------------|
| 复选框 (CheckBox)        | 创建一个复选框               |
| 组合框 (ComboBox)        | 创建一个组合框               |
| 命令按钮 (Command-Button) | 创建一个单一的命令按钮           |
| 编辑框 (EditBox)         | 创建一个编辑框               |
| 图像 (Image)            | 创建一个显示.bmp 文件的图像控件    |
| 标签 (Label)            | 创建一个用于显示正文内容的标号       |
| 线条 (Line)             | 创建一个能够显示水平线、垂直线或斜线的控件 |
| 列表框 (ListBox)         | 创建一个列表框               |
| 选项按钮 (Option-Button)  | 创建一个单一的选项按钮           |
| 形状 (Shape)            | 创建一个显示方框、圆或者椭圆的形状控件   |
| 微调按钮 (Spinner)        | 创建一个微调按钮              |
| 文本框 (TextBox)         | 创建一个文本框               |
| 计时器 (Timer)           | 创建一个能够有规律地执行代码的计时器    |

6.1.3 Visual FoxPro中的对象

1. 基本概念

在 OOP 中，对象是构成程序的基本单位和运行实体。在面向对象程序设计中，现实世界的事物均可抽象为对象，例如表单上的命令按钮是对象，表单本身也是对象。在 Visual FoxPro 中，对象又分为控件和容器两种。

控件对象是表单上显示数据和执行操作的基本对象。容器对象是可以容纳其他对象的对象。表 6.3 列出了 Visual FoxPro 的容器及其可能包含的对象。

在表单控件工具栏上的按钮中，有的能创建控件，例如命令按钮、文本框和列表框等；有的能创建容器，如命令按钮组、表格、页框等。

表 6.3 容器包含的对象

| 容器                           | 包含的对象                        |
|------------------------------|------------------------------|
| 表单集 (FormSet)                | 表单、工具栏                       |
| 表单 (Form)                    | 页框、表格，任何控件                   |
| 页框 (PageFrame)               | 页面                           |
| 页面 (Page)                    | 表格，任何控件                      |
| 表格 (Grid)                    | 表格列                          |
| 表格列 (Grid Column)            | 标头对象，除表单、表单集、工具栏、计时器和列对象外的对象 |
| 选项按钮组 (Option Button Group)  | 选项按钮                         |
| 命令按钮组 (Command Button Group) | 命令按钮                         |
| 工具栏 (ToolBar)                | 任何控件，页框，容器                   |
| 容器 (Container)               | 任何控件                         |

任何对象都具有自己的特征和行为。对象的特征由它的各种属性来描绘，对象的行为则由它的事件和方法来表达。

## 2. 对象的属性

对象的属性用来表示它的特征，以命令按钮为例，其位置、大小、颜色及该按钮上是显示文字还是图形等状态，都可用属性来表示。例如，命令按钮控件的常用属性设置如表 6.4 所示。

表 6.4 命令按钮的常用属性

| 属性        | 说明           |
|-----------|--------------|
| Name      | 命令按钮的名称，编程时用 |
| Caption   | 命令按钮上显示的文本   |
| FontName  | 命令按钮上文本的字体   |
| FontSize  | 命令按钮上文本的尺寸   |
| ForeColor | 命令按钮上文本的颜色   |
| BackColor | 命令按钮上文本的背景   |
| Top       | 命令按钮顶部的位置    |
| Height    | 命令按钮的高度      |
| Width     | 命令按钮的宽度      |

在创建对象后，必须为对象设置相应的属性，描述对象的状态。设置属性有如下几种方式。

### 1) 通过代码方式设置属性

设置对象属性的代码语法如下：

```
Parent.Object.Property = Value
```

其中，Parent 为父类名，Object 为当前对象名，Property 为属性名，Value 为对象设置的属性值。

例如，给一个表单对象“Form1”设置属性值。我们可以采用如下方式：

```
Form1.Caption="系统表单"  
Form1.BackColor=RGB(128,128,0)  
Form1.Name="form1"  
Form1.Left=30  
Form1.Top=20  
Form1.Height=250  
Form1.Width=450
```

由于每个对象都可以有多个属性，就表单对象来说，大约就有 100 个属性，进行对象属性设置时写出全部路径的输入量太大。所以 Visual FoxPro 系统还提供了一种可以对多个属性进行设置的方式，格式如下：

```
WITH <路径>  
  <属性值表>  
ENDWITH
```

可以根据这种结构来简化刚才表单的属性设置过程，例如：

```
WITH Form1
    .Caption = "系统表单"
    .BackColor = RGB(128, 128, 0)
    .Name="Form1 "
    .Left=30
    .Top=20
    .Height=250
    .Width=450
ENDWITH
```

2) 通过【属性】窗口设置属性

【属性】窗口由对象列表框、选项卡、属性设置框、属性列表框几部分组成，如图6.1所示。

① 对象列表框显示所有对象的名称，从列表中可以选选择要编辑修改的表单或控件对象，如图 6.2 所示。



图 6.1 【属性】窗口



图 6.2 对象列表框

② 选项卡由【全部】、【数据】、【方法程序】、【布局】和【其他】5部分组成。

- 【全部】：用于显示选定对象的全部属性、事件和方法代码。
- 【数据】：显示对象的数据属性。
- 【方法程序】：显示对象的方法。
- 【布局】：显示对象的外观属性。
- 【其他】：用户自定义对象属性。

③ 属性设置框可以设置或更新所选定属性列表的属性值，其中：

- ✓：接受按钮，表示确认所更改的属性值。
- ✕：取消按钮，表示取消更改，返回原来的属性值。
- $f_x$ ：函数按钮，表示可以打开一个表达式生成器来生成一个表达式。

### 3. 对象的引用

在面向对象的程序设计中常常需要引用对象，或引用对象的属性、事件与调用方法。

#### 1) 绝对引用

对象的绝对引用是指在指明对象时使用完整的容器层次来进行描述。例如在表单 Form1 中，要想指定表单中的命令按钮 Command1，可以这样引用：

```
Form1.Command1
```

#### 2) 相对引用

相对引用是为了快速引用对象。引用对象的属性或关键字如表 6.5 所示。

表 6.5 引用对象的关键字

| 属性或关键字      | 引用          | 语法格式                   | 示例（命令）                              |
|-------------|-------------|------------------------|-------------------------------------|
| Parent      | 当前对象的直接容器对象 | This.Parent            | This.Parent.Caption="OK"            |
| This        | 当前对象        | This.Objectname        | This.Caption="OK"                   |
| Thisform    | 当前对象所在的表单   | Thisform.Objectname    | Thisform.Cmd1.Caption="OK"          |
| Thisformset | 当前对象所在的表单集  | Thisformset.Objectname | Thisformset.Form1.Cmd1.Caption="OK" |

对于相对引用，通常按照以下引用关键字开头：

- Thisformset：表示当前表单集
- Thisform：表示当前表单
- This：表示当前对象
- Parent：当前对象的直接容器对象

相对引用的常用格式即引用关键字后跟一个点号，再写出被引用对象或者对象的属性、事件或方法，如下所示：

```
This.Caption      && 本对象（表单或控件）的 Caption 属性
Thisform.Cls      && 本表单的 Cls 方法，清除表单中的图形和文本
```

允许出现多级引用，但要逐级引用，例如：

```
Thisform.Command1.Caption      && 本表单的 Command1 命令按钮的 Caption 属性
This.Command1.Click            && 本对象的 Command1 命令按钮的 Click 事件
```

对于控件，也可引用包含它的容器，具体格式如下：

```
Control.Parent
```

其中 Control 表示控件，Parent 表示容器。例如 This.Parent.Command1.Caption，表示引用本对象的容器的 Command1 命令按钮的 Caption 属性。

## 6.1.4 Visual FoxPro中的事件和事件过程

### 1. 事件

事件泛指由用户或系统触发的一个特定的操作。例如，若用鼠标单击命令按钮，将会触发一个 Click 事件。一个对象可以有多个事件，但每个事件都是由系统预先规定的。一个事件对应于一个程序，称为事件过程。表 6.6 列出了 Visual FoxPro 常见的部分事件。

表 6.6 Visual FoxPro 常见的部分事件

| 事件       | 触发时机    | 事件        | 触发时机       |
|----------|---------|-----------|------------|
| Load     | 创建对象前   | MouseUp   | 释放鼠标键时     |
| Init     | 创建对象时   | MouseDown | 按下鼠标键时     |
| Activate | 对象激活时   | KeyPress  | 按下并释放某键盘键时 |
| GotFocus | 对象得到焦点时 | Valid     | 对象失去焦点前    |
| Click    | 单击鼠标左键时 | LostFocus | 对象失去焦点时    |
| DblClick | 双击鼠标左键时 | Unload    | 释放对象时      |

2. 事件驱动工作方式

事件一旦被触发，系统马上去执行与该事件对应的过程。待事件过程执行完毕后，系统又处于等待某事件发生的状态，这种程序执行方式明显不同于面向过程的程序设计，称为应用程序的事件驱动工作方式。

由上可知，事件包括事件过程和事件触发方式两方面。事件过程的代码应该事先编写好。事件触发方式可细分为 3 种：

- 由用户触发，例如单击命令按钮事件。
- 由系统触发，例如计时器事件，将自动按设定的时间间隔发生。
- 由代码引发，例如用代码来调用事件过程。

3. 为事件（或事件过程）编写代码

编写代码先要打开代码编辑窗口，打开某对象代码编辑窗口的方法有多种：

- 双击该对象
- 选定该对象的快捷菜单中的代码命令
- 选定显示菜单的代码命令

如果没有为对象的某些事件编写程序代码，那么当事件发生时系统将不会发生任何操作。如没给命令按钮的 Click 事件编写程序代码，用户即使单击该按钮，也不会产生任何操作。

4. 方法

方法是 Visual FoxPro 为对象内定的通用过程，能使对象执行一个操作。方法的过程代码由 Visual FoxPro 定义，对用户是不可见的。下面给出两个示例。

1) Cls方法

Cls 方法用于清除表单中的图形和文本，格式如下：

```
Object.Cls
```

其中的前缀 Object 表明方法的所有者，例如某个指定的表单。Cls 是方法名，相当于过程名。

2) Refresh方法

Refresh 方法可重画表单或控件，并刷新所有的值，格式如下：



```
[Form.]Object.Refresh
```

表单的 **Refresh** 方法除可在事件代码中调用外，当移动表的记录指针时，Visual FoxPro 会自动调用它，并将表单所含控件的 **Refresh** 方法全都执行一遍。

尽管方法的过程代码不可见，但还是可以修改的。但要注意，用户在代码编辑窗口写入的代码相当于为该方法增加了功能，而 Visual FoxPro 为该方法定义的原有功能并不清除。打开代码编辑窗口的方法则与事件相同。

属性或方法的设置与修改，可分在设计（交互方式操作）和运行（执行代码）两个阶段进行。对于某个属性或方法，需要了解允许在哪个阶段进行。“设计时可用”表示可通过交互操作进行设置，“运行时可用”表示可由代码来实现。一般在属性列表中显示的属性在设计时均可更改，而有些属性，如 **Caption** 在设计时和运行时均可修改。

由以上内容可知，对象的属性、事件和方法数量较多，而且一个应用程序将会包含多个对象。但是用户不必担心，因为多数属性、事件和方法不需要用户设置，只要使用默认值便可以了。

## 5. 方法的调用

一般在事件代码中调用方法，调用时必须遵循对象引用规则。例如，画图程序示例中的 **Circle** 方法，其格式如下：

```
Object.Circle (nRadius[, nXCoord, nYCoord[, nAspect]])
```

其中：

- **Object** 表示指定的表单。
- **nRadius** 表示半径。
- **nXCoord**、**nYCoord** 分别表示圆心的横坐标和纵坐标。
- **nAspect** 表示圆的纵横尺寸比，取 1.0（默认值）时产生一个标准圆，大于 1.0 产生一个垂直椭圆，小于 1.0 产生一个水平椭圆。
- **Object** 的 **ScaleMode** 属性决定度量单位。

从 **Circle** 方法的格式可知，这是带参数的方法，调用时须给出参数的实际值。例如：

```
Thisform.Circle (50, 100, 150, 1.5)
```

## 练习与操作

练习 1 比较对象和属性的特性，以及程序代码和事件的特性。

练习 2 指出容器类和控件类的不同。

练习 3 回答并完成下列问题：

- (1) 类的特征是什么？
- (2) 属性窗口的选项卡有哪些？各起什么作用？
- (3) 对象的引用有哪几种？各是什么？举例说明。
- (4) 指出 Visual FoxPro 常见的部分事件是什么？
- (5) Visual FoxPro 中的方法调用方式是什么？

## 6.2 表单的创建

在 Visual FoxPro 中,可以使用表单向导、表单设计器和表单生成器 3 种方法来创建表单。无论使用哪种方法创建并保存表单,都将产生两个文件,一个是“\*.scx”表单文件,另一个是“\*.sct”的表单备注文件。

### 6.2.1 使用表单向导创建表单

#### 1. 创建单表表单

可以使用表单向导创建一个教师基本情况的表单,操作步骤如下。

(1) 在【项目管理器】窗口中,选择【文档】选项卡,并单击其中的【表单】选项,然后单击【新建】按钮,出现【新建表单】对话框,如图 6.3 所示。

(2) 在【新建表单】对话框中,单击【表单向导】按钮,进入【向导选取】对话框。【向导选取】对话框中列出了系统提供的两种向导:表单向导和一对多表单向导,如图 6.4 所示。这里选取【表单向导】,然后单击【确定】按钮,这时启动【表单向导】对话框,屏幕显示表单向导中的“步骤 1—字段选取”对话框。



图 6.3 【新建表单】对话框



图 6.4 【向导选取】对话框

(3) 步骤 1—字段选取。在【表单向导】对话框中,选取数据库中的“教师表”,【可用字段】列表框列出了选定表中的所有字段,单击所需字段,然后单击向右箭头按钮,将它添加到【选定字段】列表框中。本例选取“教师表”中的所有字段,如图 6.5 所示。

(4) 步骤 2—选择表单样式。在【表单向导】对话框中,向导提供了 9 种表单样式供用户选择,这 9 种样式分别是:标准式、凹陷式、阴影式、边框式、浮雕式、新奇式、石墙式、亚麻式和彩色式。当用户选定一种样式后,在对话框左上角的放大镜中将显示该样式的效果。不同的按钮类型决定了表单中按钮的样式,这时提供了 4 种方式供用户选择,它们分别是“文本按钮”、“图片按钮”、“无按钮”和“定制”。这里选取表单的样式为“彩色式”,按钮类型为“文本按钮”,如图 6.6 所示。确定后单击【下一步】按钮。

(5) 步骤 3—排序次序。用来确定表单中记录的排列顺序,它的设置方法与前面学习过的查询和视图向导中的完全一样。如果选定的字段包含有重复值,则最多选择 3 个字段来进行排序,也可以选取一个索引标识来排序记录。本例选取“教师编号”字段,按升序排序表单记录,如图 6.7 示。

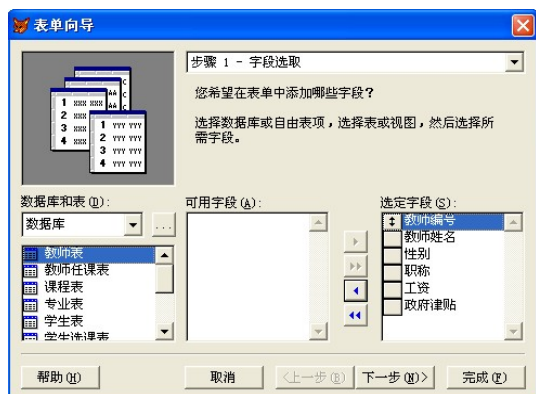


图 6.5 步骤 1—字段选取

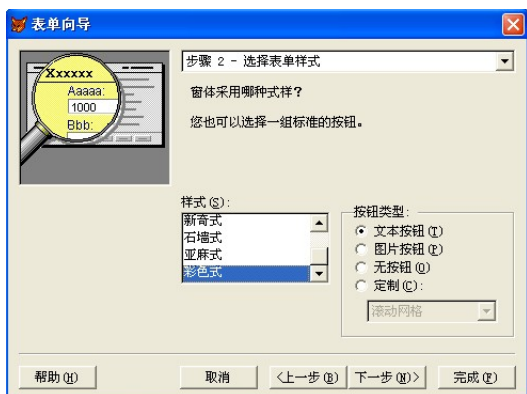


图 6.6 步骤 2—选择表单样式

(6) 步骤 4—完成。这里向导为用户提供了三种表单保存方式，可以根据自己的需要选择一种方式来保存单表。在单击【完成】按钮前，用户可以先单击【预览】按钮，观察以上所创建的表单是否满意。如果对设计不满意，可单击【上一步】按钮进行修改；如果对设计满意，可单击【完成】按钮，如图 6.8 所示。

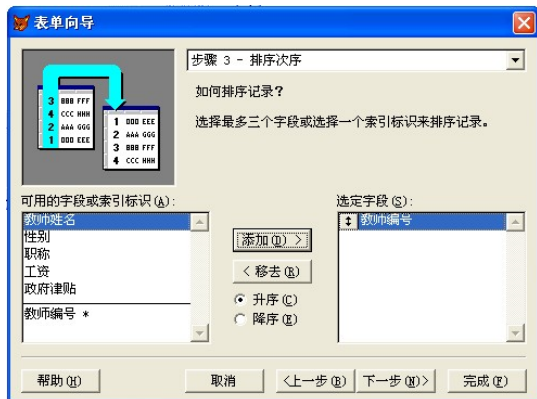


图 6.7 步骤 3—排序次序



图 6.8 步骤 4—完成

在随后弹出的【另存为】对话框中为新创建的表单输入一个文件名“教师”，保存该文件，如图 6.9 所示。

**提示：**在“步骤 4—完成”对话框中，要求键入表单的标题，这里表单的标题与保存表单输入的文件名是不一样的。表单标题显示在表单标题栏中，而表单文件名是保存表单文件的，两者是不同的。

在保存好新创建的表单后，在【项目管理器】窗口的【文档】选项卡中，展开【表单】选项，选定新创建的“教师”表单文件，并单击运行按钮，运行结果如图 6.10 所示。

从运行结果中可以看出，表单中除了显示所有的字段之外，还创建了多个功能按钮，这些按钮由向导自动提供，表单功能按钮及其含义如表 6.7 所示。

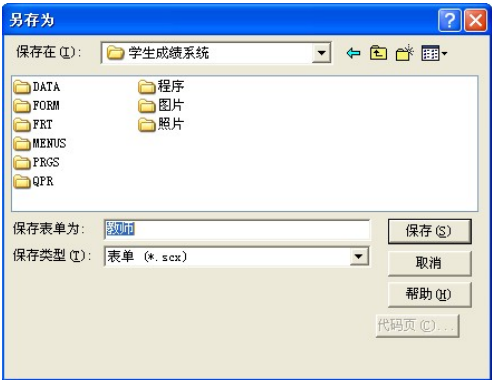


图 6.9 【另存为】对话框



图 6.10 创建的“教师”表单

表 6.7 表单的功能按钮及其含义

| 按钮名称 | 含义                  |
|------|---------------------|
| 第一个  | 移到第一条记录             |
| 前一个  | 移到上一条记录             |
| 下一个  | 移到下一条记录             |
| 最后一个 | 移到最后一条记录            |
| 查找   | 打开【搜索】对话框，查找满足条件的记录 |
| 打印   | 打印记录                |
| 添加   | 在数据表的末尾添加一条记录       |
| 编辑   | 编辑当前记录              |
| 删除   | 删除当前记录              |
| 退出   | 关闭并退出运行的表单          |

2. 创建一对多表单

前面创建的表单是基于一个表或视图的简单表单。在实际应用中，表与表之间往往存在某种关系，其中最常见的一对多的表间关系，即父表中的一条记录在子表中有多条记录和它相对应，它们之间通过某一字段相连接。

下面创建一个“教师表”和“教师任课表”之间一对多的表单。具体操作步骤如下。

(1) 在【项目管理器】窗口中，选择【文档】选项卡，单击【表单】选项，然后再单击【新建】按钮，在弹出的【新建表单】对话框中，单击【表单向导】按钮，进入【向导选取】对话框。

(2) 在【向导选取】对话框中选取【一对多表单向导】，单击【确定】按钮，这时出现一对多表单向导的第一个对话框——“步骤 1-从父表中选定字段”。

(3) 步骤 1-从父表中选定字段。选择父表“教师表”，将表中“教师编号”、“教师姓名”、“职称”3 个字段添加到【选定字段】列表框中。父表字段选定以后，单击【下一步】按钮，出现“步骤 2-从子表中选定字段”对话框，如图 6.11 所示。

(4) 步骤 2-从子表中选定字段。选择子表“教师任课表”，将全部字段添加到【选定字段】列表框中。选定子表字段以后，单击【下一步】按钮，出现“步骤 3-建立表之间的关系”对话框，如图 6.12 所示。

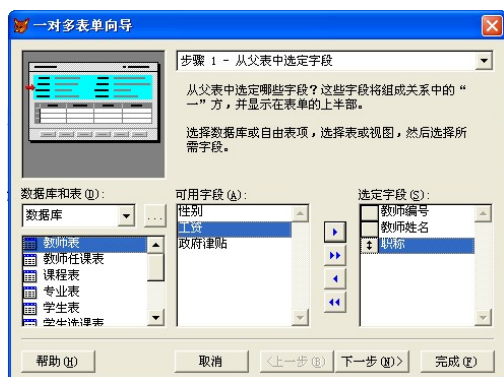


图 6.11 步骤 1—从父表中选定字段

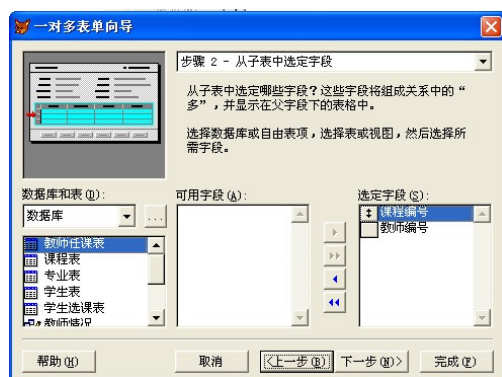


图 6.12 步骤 2—从子表中选定字段

(5) 步骤 3—建立表之间的关系。在此建立父表和子表的关系。一般选择父表中的主关键字段和子表中的某一字段相关联，这两个字段一般具有相同的名字。运行时从子表中检索与父表主关键字段相匹配的所有记录，显示在数据表格中。本例分别选取“教师表”和“教师任课表”中的“教师编号”字段，如图 6.13 所示。

(6) 步骤 4—选择表单样式。在该对话框中选择“浮雕式”，如图 6.14 所示。



图 6.13 步骤 3—建立表之间的关系



图 6.14 步骤 4—选择表单样式

(7) 步骤 5—排序次序。选取编号字段，并按升序排序，如图 6.15 所示。

(8) 步骤 6—完成。输入表单标题，确定表单存储去向，单击【完成】按钮，如图 6.16 所示。

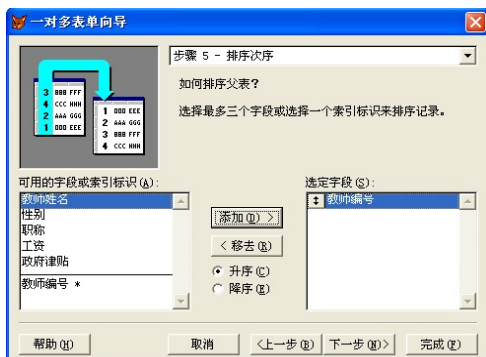


图 6.15 步骤 5—排序次序



图 6.16 步骤 6—完成

在【项目管理器】窗口中，选择【文档】选项卡中新创建的表单，单击运行按钮，可以查看一下运行结果，如图 6.17 所示。



图 6.17 新创建的表单的运行结果

从运行结果可以看到，上半部分显示的是父表“教师表”中的记录，下半部分显示了子表“教师任课表”中与父表相匹配的记录。当父表中的记录变化时，子表中的记录也相应随之变化。同时利用定位按钮来编辑父表和子表中的记录也是非常方便的。

## 6.2.2 使用表单设计器创建表单

由向导创建的表单格式单一，并且定位按钮统一。有时用户希望根据自己的需要设计表单，相应的控件和样式由自己设定，从而生成一个更完美、功能更强大的表单。下面介绍如何使用表单设计器来创建表单。

### 1. 创建快速表单

使用表单生成器可以快速创建表单，用户只需要选取字段和表单样式就可快速创建一个表单。在使用表单生成器之前，必须先启动表单设计器，然后再创建表单。

使用表单生成器快速创建基于“教师表”的一个表单，具体操作步骤如下。

#### 1) 启动表单设计器

在【项目管理器】窗口中，选择【文档】选项卡的【表单】选项，单击【新建】按钮，在弹出的【新建表单】对话框中，单击【新建表单】按钮，进入【表单设计器】窗口，显示空白表单，如图 6.18 所示。【表单设计器】窗口中还显示了一个【表单控件】工具栏，它主要用于在设计表单时添加各种控件。

#### 2) 启动表单生成器

在系统菜单中，选择【表单】菜单中的【快速表单】命令，或单击表单设计工具栏中的按钮，即可启动表单生成器，如图 6.19 所示。

#### 3) 字段选取

在【表单生成器】对话框的【字段选取】选项卡中选择数据库和表，本例选取“教师表”，并将它的部分字段添加到【选定字段】列表框中，如图 6.20 所示。



#### 4) 样式选取

在【样式】选项卡中选择适当的样式，这里选取“边框式”，如图 6.21 所示。

在选定字段和样式后，单击【确定】按钮，这时系统将快速生成表单，表单中的各个字段都将生成一个标签，其中字符型、数值型、日期型等字段生成一个文本框，逻辑型字段生成一个复选框，备注型字段生成一个编辑框，通用型字段生成一个 OLE 绑定控件。

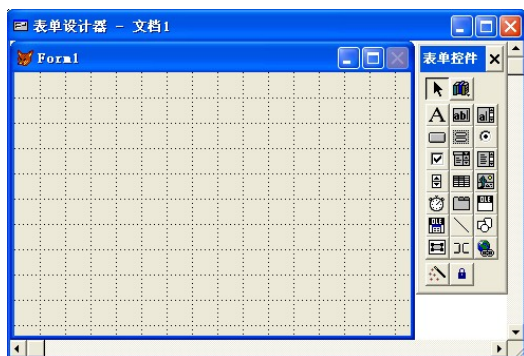


图 6.18 【表单设计器】窗口

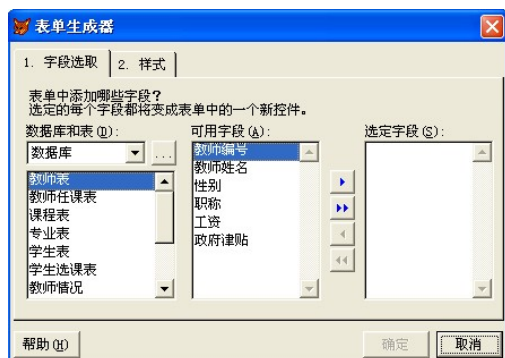


图 6.19 【表单生成器】对话框

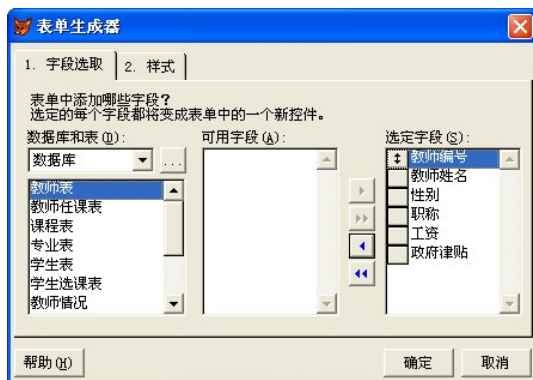


图 6.20 字段选取

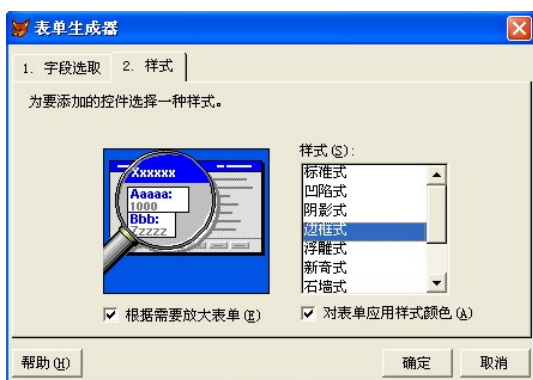


图 6.21 样式选取

#### 5) 保存表单

关闭【表单设计器】窗口，并把表单保存在“教师 1.scx”文件中。

浏览刚刚创建的表单文件“教师 1.scx”的运行结果，如图 6.22 所示。在运行的结果中可以编辑修改表单中的记录。但快速生成的表单形式单一，缺少各种定位按钮，操作非常不便，因此可以使用表单设计器来弥补它的不足。



图 6.22 表单运行结果

### 2. 使用表单设计器创建表单

使用表单向导或表单生成器能快速、方便地创建表单，设计过程很简单，但这种方法并不灵活，所创建表单的外观及功能也受到了限制。使用表单设计器，用户可以按照自己的要

求，在一张空白表单上设计出外表美观、功能齐全实用的表单。使用表单设计器，不仅可以创建基于一个表或视图的表单，还可以创建基于多个表或视图的表单（这些表或视图之间必须建立一种关联）。

利用“教师表”和“教师任课表”设计一个表单，表单中含有两个表中的部分字段，具体操作步骤如下。

### 1) 启动表单设计器

具体操作同上例。

### 2) 设置数据环境

在【表单设计器】窗口中，有一个标题为 Form1 的空白表单（其大小可以改变）。单击【显示】菜单的【数据环境】命令或表单设计工具栏中的数据环境按钮，系统启动【数据环境设计器】窗口，并显示【添加表或视图】对话框，如图 6.23 和图 6.24 所示。

如果【添加表或视图】对话框已关闭，单击【数据环境】菜单中的【添加】命令可以将它打开。

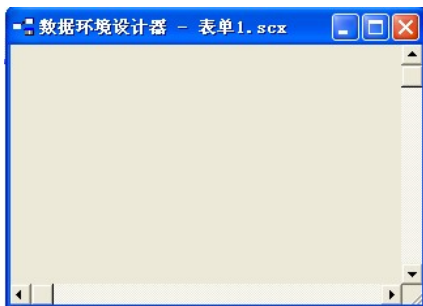


图 6.23 【数据环境设计器】窗口

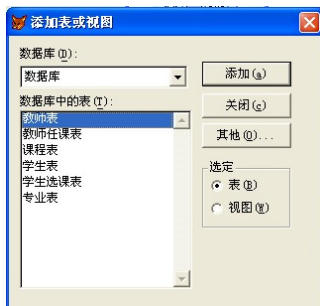


图 6.24 【添加表或视图】对话框

### 3) 建立表的关联

将数据库中的“教师表”和“教师任课表”分别添加到【数据环境设计器】窗口中，并设置两个表之间的关联，将“教师表”中的主关键字字段“教师编号”拖到“教师任课表”中的“教师编号”字段上。

### 4) 添加字段

将【数据环境设计器】窗口中的相关字段拖到【表单设计器】窗口的空白表格中。在添加字段的过程中，可以一次拖动多个字段（按住 Ctrl 键，依次单击要拖动的字段，再把字段拖到表单相应的位置上），这些字段内容将以表格的形式显示出来。对于拖到表单中的字段，单击鼠标右键，可以对它进行剪切或复制等操作。

如前所述，添加到表单中的字段都有一个标签，它显示字段的名称。如果添加的是字符型、数值型、日期型等字段，系统生成一个文本框，对于逻辑型字段生成一个复选框，对于备注型字段生成一个编辑框，对于通用型字段则生成一个 OLE 绑定控件。这种字段映像关系是系统默认的关系，用户可以通过【工具】菜单的【选项】命令进行操作。选择【字段映像】选项卡，可以设置映像字的类型、改变这种映像关系，从而创建指定类型的控件。

利用表单设计器，还可以修改和完善使用表单向导所创建的表单。



## 6.2.3 表单控件

### 1. 表单控件

控件是表单中用于显示数据、执行操作命令或修饰表单的一种对象。使用表单控件工具栏可以在表单上创建控件，【表单控件】工具栏中包括各种控件按钮，如图6.25所示。主要包括标签、文本框、组合框、列表框、编辑框、复选框、命令按钮、图像控件、OLE容器控件、OLE绑定控件及线条等。

打开【表单设计器】窗口后，单击【显示】菜单的【表单控件工具栏】命令，或在表单设计工具栏中单击表单控件按钮，从而打开或关闭【表单控件】工具栏。表单控件包括以下三类。

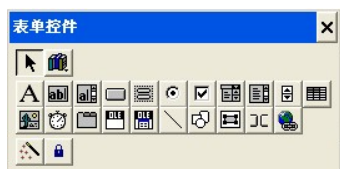


图 6.25 【表单控件】工具栏

① 常用控件。【表单设计器】窗口的【表单控件】工具栏显示的便是常用控件。这些常用控件的功能和使用方法，将在下面进行详细介绍。

② ActiveX 控件。ActiveX 控件是 OLE 自定义控件，通常用于 32 位的开发工具和平台，功能强大，应用也较复杂。

③ 自定义控件。读者还可以自己定义控件，可以选择一个可视类库作为控件添加到工具栏中。【表单控件】工具栏中各控件的功能如表 6.8 所示。

表 6.8 【表单控件】工具栏中的各种控件及其功能

| 序号 | 表单控件按钮名称 | 快捷按钮图标                                                                              | 窗体控件按钮作用                      |
|----|----------|-------------------------------------------------------------------------------------|-------------------------------|
| 1  | 选定对象     |  | 移动或改变控件大小                     |
| 2  | 查看类      |  | 选择添加可视类库                      |
| 3  | 标签       |  | 创建一个标签控件                      |
| 4  | 文本框      |  | 创建一个文本框控件，用于单行文本              |
| 5  | 编辑框      |  | 创建一个编辑框控件，用于多行文本              |
| 6  | 命令按钮     |  | 创建一个执行命令的按钮                   |
| 7  | 命令按钮组    |  | 创建一个包含多个执行命令的按钮组              |
| 8  | 选项按钮组    |  | 创建一个包含多个选项的按钮组                |
| 9  | 复选框      |  | 创建一个供选择开/关状态的复选框控件            |
| 10 | 组合框      |  | 创建一个下拉式列表框或组合框，供选择或输入数值       |
| 11 | 列表框      |  | 创建一个可上下滚动的列表框                 |
| 12 | 微调按钮     |  | 创建一个指定数值范围内的微调按钮              |
| 13 | 表格       |  | 创建一个显示数据的表格                   |
| 14 | 图像       |  | 用来向窗体中加载具有“对象链接嵌入”功能的图像、声音等数据 |
| 15 | 计时器      |  | 创建一个在指定时间或间隔后发生的事件            |
| 16 | 页框       |  | 创建一个包含多个页面的控件                 |

(续表)

| 序号 | 表单控件按钮名称 | 快捷按钮图标                                                                            | 窗体控件按钮作用                      |
|----|----------|-----------------------------------------------------------------------------------|-------------------------------|
| 17 | 未绑定对象框   |  | 添加一个不随记录变化的 OLE 对象            |
| 18 | 绑定对象框    |  | 添加一个随记录变化的 OLE 对象             |
| 19 | 线条       |  | 在表单上面添加线条                     |
| 20 | 形状       |  | 在表单上面添加各种形状的图形, 如矩形、圆角矩形、椭圆或圆 |
| 21 | 容器       |  | 在表单上添加一个容器控件                  |
| 22 | 分隔符      |  | 在工具栏的控件之间加上分隔符                |
| 23 | 超链接      |  | 创建超链接对象                       |
| 24 | 控件向导     |  | 添加控件时与生成器相关联, 从而可以自动打开生成器     |
| 25 | 按钮锁定     |  | 添加多个同类型的控件时锁定该控件, 从而不需要多次选择   |

**提示:** 在使用【表单控件】工具栏添加控件时, 如果按下生成器锁定按钮, 则在添加相关控件时, 屏幕会出现相应的生成器对话框, 系统要求设置一些与控件有关的属性。如添加列表框时, 列表框生成器要求用户选择数据表和字段、显示样式、布局等内容。如果没有按下生成器锁定按钮, 则在添加控件时, 有关控件属性需要用户自己来设置, 系统不显示生成器对话框。

2. 控件的使用方法

1) 选定控件

- ① 如果要选定一个控件, 可以单击表单中的一个控件, 其四周会出现一个有 8 个黑点组成的黑框, 表示该控件被选中。
- ② 如果要选定多个控件, 可以按住 Shift 键, 同时用鼠标单击所选控件, 即可同时选定多个控件。
- ③ 如果要选定多个相邻控件, 可以用鼠标单击空白处并拖动, 在表单设计器中画出一虚线框, 即可同时选定相邻的多个控件。

2) 分组控件

有时需要将表单上的多个控件看成是一个整体, 进行统一操作。这时可以对控件进行分组。同时选定多个控件, 在系统菜单中, 选择【格式】菜单的【分组】命令, 从而使多个控件形成一个整体。取消分组时, 只需选择【格式】菜单中的【取消分组】命令即可。

3) 移动控件

选定控件后, 单击鼠标左键并拖动, 可将选定的控件移动到表单中的任何位置。

4) 改变控件大小

将控件添加到表单上时, 其大小是固定的。用户可以根据自己的需要随意改变控件的大小。选定某一控件, 让鼠标指向某一黑点, 使其变成左右箭头、上下箭头或上下左右箭头, 并进行拖动, 可改变控件的宽度、高度或同时改变其高度和宽度。

### 5) 删除控件

选定要删除的控件，直接按下 **Delete** 键即可删除。或在系统菜单中，选择【编辑】菜单中的【清除】命令，即可删除选定的控件。

## 练习与操作

**练习 1** 以“学生表”为数据源，利用向导建立表单：学生情况一览表.scx。参考步骤如下。

- (1) 单击菜单栏上的【文件】→【新建】命令，在【新建】对话框中选择【表单】选项，单击【向导】按钮。
- (2) 在【向导选取】对话框中选择【表单向导】，单击【确定】按钮。
- (3) 在【表单向导】对话框的【数据库和表】列表框中选择想要的“学生表”，然后从【可用字段】列表框中将需要的字段（学号，姓名，性别，出生日期，入学成绩）移到【选定字段】列表框中，单击【下一步】按钮。
- (4) 选定一种样式——浮雕式，单击【下一步】按钮。
- (5) 从【可用的字段或索引标识】列表框中把“学号”字段添加到【选定字段】列表框中，选择“升序”。
- (6) 输入表单标题：“学生情况一览表”，选择【保存并运行表单】，单击【完成】按钮。结果如图 6.26 所示。

图 6.26 “学生情况一览表”表单

**练习 2** 创建“学生表”和“学生选课表”之间一对多的表单：学生选课成绩一览表.scx。参考步骤如下。

- (1) 单击菜单栏上的【文件】→【新建】命令，在【新建】对话框中选择【表单】选项，单击【向导】按钮。
- (2) 在【向导选取】对话框中选择【一对多表单向导】，单击【确定】按钮。
- (3) 在【表单向导】对话框中的【数据库和表】列表框中选择想要的父表“学生表”，然后从【可用字段】列表框中将需要的父表字段（学号，姓名，性别）移到【选定字段】列表框中，单击【下一步】按钮。
- (4) 在【表单向导】对话框的步骤 2 中，在【数据库和表】列表框中选择想要的子表“学生选课表”，然后从【可用字段】列表框中将需要的子表的所有字段移到【选定字段】列表框中，单击【下一步】按钮。
- (5) 在此建立父表和子表的关联。将“学生表”和“学生选课表”中的“学号”字段进行关联。
- (6) 选定一种样式——浮雕式，单击【下一步】按钮。

- (7) 从【可用的字段或索引标识】列表框中把“学号”字段添加到【选定字段】列表框中，选择“升序”，单击【下一步】按钮。
- (8) 输入表单标题：“学生选课成绩一览表”，选择【保存并运行表单】，单击【完成】按钮。结果如图 6.27 所示。



图 6.27 “学生选课成绩一览表”表单

- 练习 3** 利用“学生成绩”视图设计一个表单，表单中含有学生选课成绩的部分字段。参考步骤如下。
- (1) 单击菜单栏上的【文件】→【新建】按钮，在【新建】对话框中选择【表单】选项，单击【新建文件】按钮，打开【表单设计器】窗口。在【表单设计器】窗口中，有一个标题为 Form1 的空白表单（其大小可以改变）。
- (2) 设置数据环境。单击【显示】菜单中的【数据环境】命令或表单设计工具栏中的数据环境按钮，系统启动【数据环境设计器】窗口，并显示【添加表或视图】对话框，单击【视图】选项后选择“学生成绩”视图。
- (3) 添加字段。将【数据环境设计器】窗口中的所有字段拖到【表单设计器】窗口的空白表格中。在添加字段的过程中，可以一次拖动多个字段（按住 Ctrl 键，依次单击要拖动的字段，再把字段拖到表单相应的位置上），这些字段内容以表格的形式显示出来。
- (4) 从表单控件工具栏中拖入 1 个标签、4 个命令按钮，并调整其大小和位置。
- (5) 设置各控件的属性如表 6.9 所示。

表 6.9 各控件的属性

| 控件     | 属性       | 属性值       |
|--------|----------|-----------|
| 标签 1   | Caption  | 学生选课成绩一览表 |
|        | Name     | Label1    |
|        | FontSize | 22        |
| 其他标签   | 默认       | 默认        |
| 各文本框   | 默认       | 默认        |
| 命令按钮 1 | Caption  | 第一个       |
|        | Name     | Command1  |
| 命令按钮 2 | Caption  | 上一个       |
|        | Name     | Command2  |
| 命令按钮 3 | Caption  | 下一个       |
|        | Name     | Command3  |
| 命令按钮 4 | Caption  | 末一个       |
|        | Name     | Command4  |

(6) 命令按钮代码的设置。

“第一个”按钮的代码:

```
GO TOP  
Thisform.Refresh
```

“上一个”按钮的代码:

```
SKIP -1  
IF BOF()  
GO TOP  
ENDIF  
Thisform.Refresh
```

“下一个”按钮的代码:

```
SKIP  
IF EOF()  
GO BOTTOM  
ENDIF  
Thisform.Refresh
```

“末一个”按钮的代码:

```
GO BOTTOM  
Thisform.Refresh
```

“关闭”按钮的代码:

```
Thisform.Refresh
```

(7) 保存并运行表单, 结果如图 6.28 所示。

图 6.28 表单运行结果

## 6.3 常用控件的使用方法

### 6.3.1 标签、文本框和命令按钮（组）的使用

#### 1. 标签

标签是表单中应用最广泛的一种控件, 它可以单独使用, 也可以与其他控件结合使用, 描述其信息。

### 1) 功能

标签控件用于显示文本信息，为表单提供信息说明。它没有数据源，用户只能通过表单中的代码来改变标签控件中的内容，而不能直接对其内容进行交互式编辑。因此，标签控件无法作为输入信息的界面。

### 2) 常用属性

- **AutoSize**: 选择标签是否会根据标题的长度自动调整其大小，默认值为“.F.”。
- **BackColor**: 设置标签的背景颜色。
- **BackStyle**: 选择标签是否为透明的，默认值为假，即不透明。
- **Caption**: 设置标签控件显示的文本内容，最大长度为 256 个字符。
- **FontSize**: 设置标签中字体的大小。
- **ForeColor**: 设置标签中标题的颜色。
- **Visible**: 设置是否显示标签控件。
- **WordWrap**: 选择标签控件中显示的文本是否换行，默认值为“.F.”。

### 3) 示例

在空白表单上设置一个标题为“学生选课情况一览表”，操作步骤如下。

(1) 单击【表单控件】工具栏中的标签按钮，再将鼠标指针拖到表单上，单击鼠标左键，就在表单上产生了一个默认大小的标签。按下鼠标左键并拖动鼠标，可产生任意大小的标签。

(2) 在【属性】窗口的 **Caption** 属性中，输入标签内容，如“学生选课情况一览表”。使用同样的方法可以添加其他几个标签，添加的标签被选中后，就可以移动它的位置并改变它的大小。在表单上添加一标签控件，并拖动成适当的大小，如图 6.29 所示。

(3) 选定“学生选课情况一览表”标签，设置 **FontSize** 为“22”。

(4) 选择 **BackColor** 属性设置背景颜色。单击文本框右侧的按钮，打开【颜色】对话框，从中选择蓝色，单击【确定】按钮。

(5) 选择 **ForeColor** 属性设置前景颜色。单击文本框右侧的按钮，打开【颜色】对话框，从中选择白色，单击【确定】按钮。

(6) 在系统菜单中，选择【格式】菜单的【大小】选项，从其子菜单中选择【恰好容纳】命令。再选择【格式】→【对齐】→【水平居中】命令。最后结果如图 6.30 所示。



图 6.29 添加的标签控件



图 6.30 显示结果

## 2. 文本框

文本框控件用于在表单中创建一个文本框，是用来显示和编辑数据的控件，一般用它来显示一个非备注型字段值。随着记录指针的变化，文本框中显示的内容也随之变化。文本框是表单中最常用的控件之一。

### 1) 功能

① 输入/输出。文本框不仅可以输入/输出除备注类型外的各种类型的数据，还可以设置输入/输出格式。如将其 `InputMask` 属性设置为 `99.9`，则它决定了输入的数据必须小于 `100`，并且只保留一位小数。

② 编辑功能。在文本框中可以进行剪切、复制和粘贴等操作。如果设置了文本框的 `ControlSource` 属性，那么文本框中显示的内容除了保存在它的 `Value` 属性中，同时也保存在 `ControlSource` 属性指定的表字段或变量中。

③ 数据验证。在文本框的 `Valid` 事件中写进相应的检验代码，可以检验文本框中的数据是否符合规则。如果不符合规则，系统会给出提示信息，并返回值 `“.F.”`。

④ 控制显示。通常使用密码来保证应用程序的安全性。为了不让用户输入的密码显示在屏幕上，可以设置其 `PasswordChar` 属性，通常将该属性的值设置为 `“*”`。这样用户在输入密码时，屏幕上只显示 `“*”`，而实际值则保存在文本框的 `Value` 属性中。

### 2) 常用属性

- `ControlSource`：设置控件数据的来源。
- `FontName`：设置文本框中字体的类型。
- `Format`：设置文本框中值的显示方式。
- `InputMask`：设置文本框中值的输入格式及范围。
- `Name`：设置文本框的名称。
- `PasswordChar`：设置文本框中显示的字符。
- `Valid`：双击选定的文本框，或在【属性】窗口选择方法代码中的 `Valid` 事件，均可打开如图 6.31 所示的窗口，在该窗口中可输入事件代码。
- `Value`：用于保存文本框中的值。

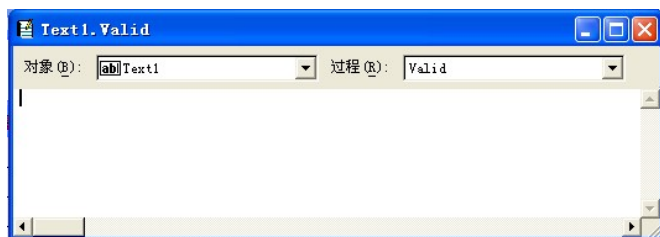


图 6.31 方法的代码输入框

### 3) 示例

在上例中所创建的表单标签之后，添加相应的文本框控件，操作步骤如下。

(1) 按下【表单控件】工具栏中的生成器锁定按钮，再按下文本框按钮。



(2) 将鼠标拖到学生学号标签右侧，单击鼠标左键，添加一个文本框 `Text1`，并启动文本框生成器。

(3) 【文本框生成器】窗口包括三个选项卡：

- **【格式】**：设计文本框格式界面，这里数据类型选择字符型。
- **【样式】**：设计文本框排列方式。
- **【值】**：选择保存文本框值的一个字段。这里选择“学生表.学号”字段。单击**【确定】**按钮，完成文本框控件的添加操作。

同样的方法可以在“姓名”和“课程编号”等字段的右侧添加文本框，并设置相应的数据类型和样式。

### 3. 命令按钮

#### 1) 功能

用于在表单上创建单个命令按钮，当单击该命令按钮时，可以触发该命令按钮的事件，执行一个特定的操作。

#### 2) 常用属性

- **Caption**：设置命令按钮的标题、如添加、编辑、保存、退出等。
- **Picture**：设置在命令按钮上显示的图形文件。如果在选择该属性的同时也选择了 **Caption** 属性，则图形在命令按钮的上半部分显示。此时命令按钮要足够大，否则图形无法全部显示出来，因为图形部分不能强占标题部分的大小。
- **DownPicture**：设置当命令按钮按下后显示的图形文件。
- **Default**：当该属性的值为“.T.”时，可以用 **Enter** 键指定该命令按钮。
- **Cancel**：当该属性的值为“.T.”时，可以用 **Esc** 键指定该命令按钮。
- **Enabled**：指定命令按钮是否有效。为了避免误操作，当表单不能执行某些操作时，可将其相应的命令按钮设置为无效。所以这是一个非常重要的属性。
- **DisablePicture**：设置当按钮无效时要显示的图形。
- **Click**：双击选定的命令按钮，将打开 **Click** 编辑窗口，用户可在此窗口中输入事件代码。

#### 3) 示例

创建两个命令按钮“显示”、“退出”，并运行该表单。操作步骤如下。

(1) 按下**【表单控件】**工具栏中的命令按钮，将鼠标指针移到表单中，单击鼠标左键，系统在表单上添加了一个命令按钮 `Command1`。随后添加另一个命令按钮 `Command2`。

(2) 在**【属性】**窗口的**【全部】**选项卡中，将 `Command1` 的 **Caption** 属性值改为“显示”，将 `Command2` 的 **Caption** 属性值改为“退出”。

(3) 双击“显示”按钮，进入 **Click Event** 属性框，在打开的 `Command1.Click` 编辑窗口中输入命令执行代码：`DO FORM d:\12\Form1.scx`。双击“退出”按钮，进入 **Click Event** 属性框，在打开的 `Command2.Click` 编辑窗口中输入命令执行代码：`Thisform.Release`。

(4) 结束代码输入并单击工具栏中的运行按钮，再单击表单中的“表单1”，则将前面建立的表单运行结果显示在屏幕上。



## 4. 命令按钮组

在使用表单向导创建表单时，系统提供了一些标准的定位、浏览、编辑等按钮，便于表单操作。读者已经注意到，本书前面创建的表单中没有定位、浏览等按钮，下面介绍创建命令按钮组控件，并将许多命令按钮设计成一组。

### 1) 功能

命令按钮组的功能是在表单上创建一组命令按钮。

### 2) 常用属性

- **ButtonCount**: 设置命令按钮组中命令按钮的个数。
- **Enabled**: 指定命令按钮或命令按钮组是否有效。如果同时设置了命令按钮组和命令按钮组中某个命令按钮的 **Enabled** 属性，且它们的属性不相同，则以命令按钮组的 **Enabled** 属性值为准。

### 3) 示例

建立一个命令按钮组，其中有 3 个命令按钮：红色、绿色和蓝色按钮，单击其中一个命令按钮显示相应的背景颜色，操作步骤如下。

(1) 打开表单 1，按下【表单控件】工具栏中的命令按钮组，将鼠标指针移到表单下方，单击鼠标左键，在表单上添加了一个命令按钮组 **Command1**、**Command2** 和 **Command3**，并启动了命令组生成器。

(2) 在【命令组生成器】窗口的【按钮】选项卡中，设置按钮数目为 3，标题分别为“红色”、“黄色”和“蓝色”。在【布局】选项卡中将按钮布局设置为水平，结果如图 6.32 所示。

(3) 设置命令按钮组的 Click 事件代码：

```
DO CASE
CASE This.Value=1
    Thisform.BackColor=RGB(255,0,0)
CASE This.Value=2
    Thisform.BackColor=RGB(0,255,0)
CASE This.Value=3
    Thisform.BackColor=RGB(0,0,255)
ENDCASE
```

(4) 保存表单，运行该表单，验证这个命令按钮组所完成的功能，如图 6.33 所示。



图 6.32 添加的控件



图 6.33 显示结果

## 6.3.2 编辑框、列表框和组合框的使用

### 1. 编辑框

编辑框与文本框类似，但其只能接受字符类型的数据。

#### 1) 功能

编辑框是编辑备注型字段的常用控件，因为它的最大容量为 2 147 483 646 个字符，而且具有 Visual FoxPro 中所有标准的编辑功能。

#### 2) 常用属性

- ScrollBars: 选择编辑框是否具有垂直滚动条。
- ReadOnly: 选择是否可以修改编辑框中的文本。
- SelLength: 设置选择文本的长度。
- SelStart: 设置选择文本的开始位置。
- SelText: 设置选定的文本。

#### 3) 示例

为上例添加一个编辑个人简历的编辑框，参考步骤如下。

(1) 在表单上添加一标签控件和一编辑框，并调整好控件的大小，如图 6.34 所示。

(2) 打开标签控件的【属性】窗口，将标签的 Caption 属性修改为“简历”，将 FontSize 属性设为“18”，并选择【格式】菜单的【大小】选项，从其子菜单中选择【恰好容纳】命令。

(3) 打开编辑框的【属性】窗口，设置编辑框的数据来源。将其 ControlSource 属性设为“简历”，FontSize 属性设为“10”。

(4) 运行表单，结果如图 6.35 所示。可以看到编辑框中显示了“学生选课情况一览表”中第一个记录的内容。



图 6.34 添加编辑框



图 6.35 运行结果

### 2. 列表框

#### 1) 功能

列表框控件用于在表单中创建列表框，以列的形式显示一系列数据供用户选择，用户也可直接输入文本。

## 2) 常用属性

- **ColumnCount**: 设置列表框的列数, 默认值为 0。
- **ControlSource**: 设置用于保存用户从列表框中选择的值的名称, 如字段、变量等。
- **MoveBar**: 选择是否在列表项左侧显示用于对列表项进行重新排列的移动按钮, 默认值为“.F.”。
- **Picture**: 指定在列表项前显示的图形文件。
- **MultiSelect**: 指定是否允许用户可以一次从列表框中选择多项, 默认值为“.F.”。

## 3) 示例

利用列表框从“专业表”中读取“专业名称”数据, 并将其显示在表单中。操作步骤如下。

(1) 在上例中所创建的表单上添加一个“专业名称”标签, 再按下列表框按钮, 在毕业学校标签右侧添加一个列表框, 并显示【列表框生成器】对话框(如果没按下生成器锁定按钮, 可将鼠标指向列表框, 单击鼠标右键, 在出现的快捷菜单中打开生成器)。

(2) 在【列表框生成器】对话框中包含下列选项卡:

- **【列表项】**: 列表框中添加数据的来源, 这里选择“专业表.专业名称”字段。
- **【样式】**: 设计列表框排列方式。
- **【布局】**: 调整列表框中数据的行宽。
- **【值】**: 对话框上方的输入框中指定数据的来源, 这里选择“专业表”。下方的输入框指定用来存储数值的表或视图的字段, 这里选择“专业名称”字段。

(3) 单击工具栏中的运行按钮, 保存表单并运行该表单, 从列表框中选择相应的专业。

列表框能够提供数据列表, 用户从列表框中选择数据, 并返回数据。可以上下滚动列表框, 操作起来很方便, 如图 6.36 所示。

## 3. 组合框

组合框兼有列表框和文本框的功能, 可以在其中输入值或从列表中选择数据项。



图 6.36 添加列表框

## 1) 功能

用于在表单中创建组合框, 提供一组数据供用户选择, 通常只有一个条目是可见的。

## 2) 常用属性

- **DisplayCount**: 指定在组合框中允许显示的最大数目。
- **ControlSource**: 设置用于保存用户从组合框中选择的值的名称, 如字段、变量等。
- **RowSource**: 指定组合框中数据的来源。
- **RowSourceType**: 指定组合框中数据的来源的类型。

- **Style:** 指定组合框是下拉组合框还是下拉列表框。
- **Text:** 返回输入到组合框中的文本框部分的文本。

### 3) 示例

在上例的基础上，建立一个组合框，利用组合框从“专业名称”中读取数据，并将其显示在表单中，观察列表框与组合框的区别。操作步骤如下。

(1) 在上例中所创建的表单上添加一个“专业名称”标签，再按下组合框按钮，在“专业名称”标签右侧添加一个组合框，并显示【组合框生成器】对话框（如果没按下生成器锁定按钮，可将鼠标指向组合框，单击鼠标右键，在出现的快捷菜单中打开生成器）。

(2) 在【组合框生成器】对话框中包含下列选项卡：

- **【列表项】:** 组合框中添加数据的来源，这里选择“专业表.专业名称”字段。
- **【样式】:** 设计组合框排列方式。
- **【布局】:** 调整组合框中数据的行宽。
- **【值】:** 对话框上方的输入框中指定数据的来源，这里选择“专业表”。下方的输入框指定用来存储数值的表或视图的字段，这里显示“专业名称”字段。

(3) 单击工具栏中的运行按钮，保存表单并运行该表单，单击组合框右侧的下拉箭头，即可从下拉列表框中选择相应的专业。

组合框能够提供数据列表，方便用户从下拉列表中选择数据。示例的运行结果如图 6.37 所示。

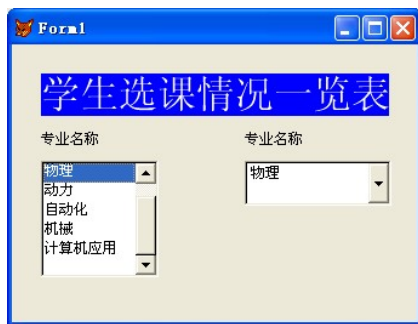


图 6.37 示例的运行结果

## 6.3.3 选项按钮组、复选框和微调按钮的使用

### 1. 选项按钮组

选项按钮组是一个包含多个单项按钮的容器，用来显示多个选项，可以从一系列的选项中选择一项，但要注意同时只能选择其中一项，且被选中的选项按钮中只显示一个圆点。

#### 1) 功能

该控件用于在表单中创建选项按钮组，用于显示多个选项，但只能选择一项。

#### 2) 常用属性

- **ButtonCount:** 指定选项组中选择按钮的个数。
- **ControlSource:** 设置选项组的控制源。
- **Value:** 指定选项按钮组中的当前选项是第几个选项。
- **Click:** 单击选项按钮组时触发事件。
- **RightClick:** 右击选项按钮组时触发事件。

### 3) 示例

在上例的基础上，建立一个选项按钮组，其中有3个选项按钮：红、绿和蓝。单击其中一个命令按钮显示标题“学生选课情况一览表”的前景颜色，操作步骤如下。

(1) 打开表单1，选择【表单控件】工具栏中的选项按钮组，将鼠标指针移到表单下方，单击鼠标左键，在表单上添加了一个选项按钮组 Option1、Option2 和 Option3，并启动了命令组生成器。表单布局如图 6.38 所示。

(2) 在【命令组生成器】窗口的【按钮】选项卡中，设置按钮数目为3，标题分别为“红”、“绿”和“蓝”。在【布局】选项卡中将按钮布局设置为水平。

(3) 设置选项按钮组的 Click 事件代码：

```
DO CASE
CASE This.Value=1
    Thisform.Label1.ForeColor=RGB(255,0,0)
CASE This.Value=2
    Thisform.Label1.ForeColor=RGB(0,255,0)
CASE This.Value=3
    Thisform.Label1.ForeColor=RGB(0,0,255)
ENDCASE
```

## 2. 复选框

复选框是只有两个逻辑值选项的控件，可以选择开关状态。复选框与选项按钮组不同，在选项组中同时只能选择一个单选项，而在复选框中可以同时选中多项。

### 1) 功能

用于在表单中创建复选框，用于显示开关选项，当选项为真（.T.）时，复选框内显示一个对钩；否则，复选框内为空白。

### 2) 常用属性

- Caption：设置命令按钮的标题。
- Value：指定复选框是否为选定状态。
- ControlSource：指定复选框的数据来源。

### 3) 示例

在上例的基础上，通过建立3个复选框的选取来决定是否给标题“学生选课情况一览表”增加下划线、斜体和粗体效果，操作步骤如下。

(1) 打开表单1，选择【表单控件】工具栏中的复选框，将鼠标指针移到表单下方，单击鼠标左键，在表单上添加3个复选框。

(2) 在3个复选框边上分别将标题设置为“粗体”、“斜体”和“下划线”。

(3) 设置3个复选框的 Click 事件代码。

复选框1：

```
IF Thisform.Check1.Value=1
```

```

        Thisform.Label1.FontBold=.T.
    ELSE
        Thisform.Label1.FontBold=.F.
    ENDIF

```

复选框 2:

```

    IF Thisform.Check2.Value=1
        Thisform.Label1.FontItalic=.T.
    ELSE
        Thisform.Label1.FontItalic=.F.
    ENDIF

```

复选框 3:

```

    IF Thisform.Check3.value=1
        Thisform.Label1.FontUnderline=.T.
    ELSE
        Thisform.Label1.FontUnderline=.F.
    ENDIF

```

(4) 保存表单，并运行该表单，验证上述控件所完成的功能，如图 6.39 所示。



图 6.38 添加各类控件



图 6.39 运行结果

### 3. 微调按钮

微调按钮控件可以用来反映字段或变量的数值变化，并将值写在相应的字段或变量中。

#### 1) 功能

微调按钮用来控制数值型数据的使用范围，并接收给定范围内的数值输入、数据调整 and 选择。利用微调按钮可以输入一个数据，也可以代替键盘输入接收一个值，还可以通过按钮在当前值的基础上进行微小的增量和减量调节。

#### 2) 常用属性

- **Increment:** 设置每次单击按钮的变化量，即每次单击向上或向下按钮时增加或减少的值。
- **KeyboardHighValue:** 设置用户能从键盘键入到微调文本框中的最高值。
- **KeyboardLowValue:** 设置用户能从键盘键入到微调文本框中的最低值。

- **SpinnerHighValue**: 设置微调按钮控件能显示的最高值, 即设置其可选数据范围的上限。
- **SpinnerLowValue**: 设置微调按钮控件能显示的最低值, 即设置其可选数据范围的下限。

### 3) 示例

颜色数字的设置范围为“0~255”, 用 3 个微调按钮分别代表“红、绿、蓝”。3 个微调按钮的 **BackColor** 属性最高应该设为 255, 最低设为 0, 并通过标签 **Label4** 显示设置结果。颜色配置函数为 **RGB(n1, n2, n3)**; 通过滚动条的 **Change** 事件实现颜色值的动态改变, 并显示到标签中。操作步骤如下。

- (1) 启动 Visual FoxPro。
- (2) 添加窗体 **Form1**。
- (3) 在窗体中添加以下控件: 4 个标签、3 个微调按钮, 如图 6.40 所示。
- (4) 各控件属性设置如表 6.10 所示。

表 6.10 各控件属性

| 控件     | 属性      | 属性值      |
|--------|---------|----------|
| 窗体     | Caption | 调色板测试    |
|        | Name    | Form1    |
| 标签 1   | Caption |          |
|        | Name    | Label1   |
| 标签 2   | Caption | 红        |
|        | Name    | Label2   |
| 标签 3   | Caption | 绿        |
|        | Name    | Label3   |
| 标签 4   | Caption | 蓝        |
|        | Name    | Label4   |
| 命令按钮 1 | Caption | 关闭       |
|        | Name    | Command1 |
| 微调按钮 1 | Name    | Spinner1 |
| 微调按钮 2 | Name    | Spinner2 |
| 微调按钮 3 | Name    | Spinner3 |

- (5) 编写代码。

设置微调按钮 **Spinner1** 的 **InteractiveChange** (交互改变) 事件代码:

```
r=This.Parent.Spinner1.Value  
g=This.Parent.Spinner2.Value  
b=This.Parent.Spinner3.Value  
This.Parent.Label1.BackColor=RGB(r,g,b)
```

设置微调按钮 **Spinner2** 的 **InteractiveChange** (交互改变) 事件代码:

```
r=This.Parent.Spinner1.Value  
g=This.Parent.Spinner2.Value  
b=This.Parent.Spinner3.Value  
This.Parent.Label1.BackColor=RGB(r,g,b)
```

设置微调按钮 Spinner3 的 InteractiveChange (交互改变) 事件代码:

```
r=This.Parent.Spinner1.Value
g=This.Parent.Spinner2.Value
b=This.Parent.Spinner3.Value
This.Parent.Label1.BackColor=RGB(r,g,b)
```

(6) 运行表单, 如图 6.41 所示。



图 6.40 添加微调按钮控件

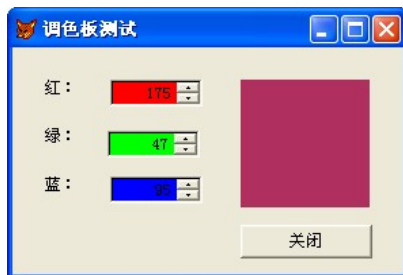


图 6.41 运行结果

## 6.3.4 表格和计时器的使用

### 1. 表格

#### 1) 功能

表格控件是表单中最常用的控件之一, 用于在表单中创建一个表格。其外观与浏览窗口类似, 按行和列显示对象。表格一般用来对指定表或视图中的记录进行维护和显示。在表格中含有列, 列中又含有列标头, 它们都有各自的属性、事件和方法。通过修改它们的属性、事件和方法, 可以指定表格中显示的内容。另外在表格的列中还可以添加文本框、复选框、微调按钮等控件, 从而使表格中数据的输入更加方便。

#### 2) 常用属性

- **ColumnCount**: 指定表格所包含的列数, 默认值为-1, 即表格中包含与它关联的表中的所有字段。
- **DataSource**: 设置在列中显示的数据源。
- **ScrollBars**: 指定表格中是否显示滚动条。其中包含:
  - 0: 表格中不显示滚动条。
  - 1: 表格中显示一个水平滚动条。
  - 2: 表格中显示一个垂直滚动条。
  - 3: 表格中显示一个水平和一个垂直滚动条。
- **ControlSource**: 设置列中要显示的数据, 通常是表中的一个字段。
- **CurrentControl**: 指定表格中的活动控件, 默认值为 Text1, 即第一个文本框。
- **Width**: 设置列的宽度。
- **Caption**: 设置列标头显示的内容。



### 3) 示例

在 Form1 表单中添加一个表格控件，表格内容是“学生选课表”中的记录。操作步骤如下。

(1) 按下【表单控件】工具栏中的生成器锁定按钮，再按下表格控件按钮。

(2) 在表单上要放置表格的地方单击鼠标左键，这时在表单上添加一个表格，同时启动【表格生成器】窗口。

(3) 在【表格生成器】窗口中包含 4 个选项卡：

- **【表格项】**：选择表格中要显示的字段名称，本例选取“学生选课表”中的全部字段。
- **【样式】**：为表格选择一种显示样式。
- **【布局】**：指定一个列标题和控件类型。
- **【关系】**：若创建一个一对多表单，需要定义父表和子表之间的关系。



| 学号        | 课程编号 | 开课时间     | 成绩 |
|-----------|------|----------|----|
| 070401032 | CC02 | 02/25/09 | 92 |
| 070401032 | CS01 | 02/23/08 | 91 |
| 070403128 | CC01 | 02/23/08 | 86 |
| 074101067 | CC01 | 02/20/09 | 67 |
| 074101067 | CC03 | 09/01/08 | 82 |
| 074202123 | CS02 | 09/01/09 | 90 |
| 080301001 | CC01 | 02/20/09 | 96 |

图 6.42 添加表格后的运行结果

(4) 单击工具栏中的运行按钮，保存表单并运行该表单，结果如图 6.42 所示。

## 2. 计时器

### 1) 功能

用于在表单中创建一个计时器，计时器会以一定的时间间隔执行事先编写的事件代码。在 Timer 事件中，放入需要重复执行的事件代码，如检索系统时钟、定时完成一些后处理等。在表单设计器中，计时器是可见的，便于设计者选择该控件，设置其属性和编写事件过程。而运行表单后，计时器就不可见了，因此它的大小和位置不会对表单的界面有任何影响。

### 2) 常用属性

- **Enabled**：设置计时器是否工作。该属性值为“.T.”时，计时器开始工作；否则，计时器将被挂起。另外，该属性也可以通过触发其他控件的事件来设置。
- **Interval**：设置两次计时器事件（Timer 事件）的时间间隔，单位为毫秒。该属性的值不要设置得太小，否则占用处理器的时间太多，会降低整个程序的性能。

### 3) 示例

在表单上添加一个计时器控件，操作步骤如下。

(1) 在表单中添加一个标签控件和一个计时器控件。

(2) 将标签控件的 Caption 属性值修改为“时间：”，FontSize 属性值设为 28，Name 属性值设为 Label1。

(3) 将计时器的 Enabled 属性值修改为“.T.”，Interval 属性值设为 600（ms）。布局结果如图 6.43 所示。

(4) 双击计时器控件，打开其事件代码窗口，编写代码：Thisform.Label1.Caption = TIME()。

(5) 运行表单，其结果如图 6.44 所示，这时标签显示的是系统时间。正如前面介绍的，标签不能在表单上交互动地编辑其文本内容，但可以通过表单中的代码改变标签控件中的内容。



图 6.43 在表单中添加计时器控件

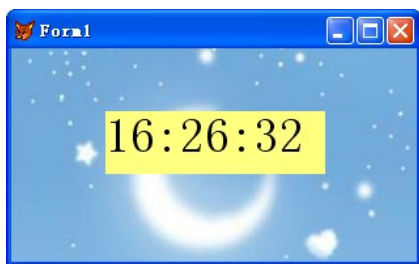


图 6.44 运行结果

## 6.3.5 OLE绑定控件和图像的使用

### 1. OLE绑定控件

#### 1) 功能

OLE 绑定控件常用来在表单上显示与通用型字段有关的 OLE 对象，显示内容随着记录的变化而变化。因此，它与数据表中的通用型字段相连接。

#### 2) 常用属性

- **ControlSource**: 设置与数据表中某一通用型字段相连接。
- **Stretch**: 设置 OLE 对象与显示区域的大小比例。它包括 3 种情况:
  - 0, 剪裁 (默认值): 超过显示区域部分的图像被剪去。
  - 1, 等比填充: OLE 对象等比例放大或缩小显示。
  - 2, 变比填充: 以显示区域为前提, 显示整个 OLE 对象。

#### 3) 示例

在“学生选课情况一览表”表单中, 添加一个 OLE 绑定控件, 使它与“学生表.照片”字段相连接。操作步骤如下。

(1) 打开“学生选课情况一览表”表单, 按下【表单控件】工具栏中的 OLE 绑定控件按钮, 在表单的右上方添加一个 OLE 绑定控件。

(2) 设置 OLE 绑定控件属性。将 **ControlSource** 属性设置为与“学生表”的通用型字段相连接的“学生表.照片”字段。将 **Stretch** 属性的值设置为 0 (剪裁)。布局结果如图 6.45 所示。

(3) 保存并运行该表单, 结果如图 6.46 所示。

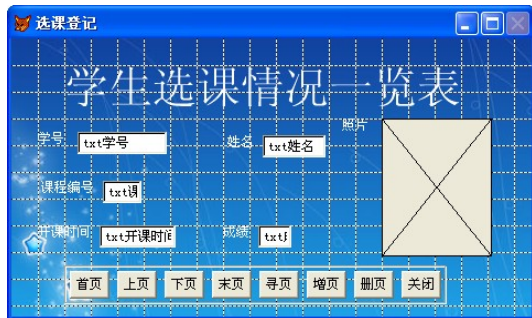


图 6.45 添加 OLE 绑定控件



图 6.46 运行结果

## 2. 图像

### 1) 功能

图像控件用于在表单上添加一个.bmp 图像文件所包含的图片，以美化表单界面的设计，但不能直接修改该图片。

### 2) 常用属性

- **Picture**: 指定该控件要显示的.bmp（或.jpg）图像文件。
- **BorderStyle**: 选择是否显示边框，默认状态为无边框。
- **Stretch**: 设置图像的填充方式，其中：
  - 0，剪裁：系统会自动地剪裁图像的大小，可能会导致图像无法全部显示出来，该选项是默认值。
  - 1，等比填充：按原图像的比例进行缩放，在不改变图像原来比例的条件下，根据控件的大小自动调整图像，使其尽量填满控件。
  - 2，变比填充：根据控件的大小，自动调整图像，但其为了填满控件，也许无法保留图像原有的比例从而使图像失真。

### 3) 示例

在表单上添加 3 个图像控件，其大小和形状如图 6.47 所示。操作步骤如下。

(1) 将这 3 个图像控件的 **Stretch** 属性分别设置为：0—剪裁，1—等比填充，2—变比填充。

(2) 设置这 3 个图像控件的 **Picture** 属性：单击输入框右侧的按钮，在【打开】对话框中选择“小熊 1.jpg”、“小熊 2.jpg”、“小熊 3.jpg” 3 个文件，可以在预览区域预览图像内容，然后单击【确定】按钮。

(3) 属性设置完后，运行表单，其结果如图 6.48 所示，从表单中可以看到 3 个图像控件由于 **Stretch** 属性值设置的不同，所显示的图像的形状也不尽相同。



图 6.47 添加图像控件



图 6.48 运行结果

## 练习与操作

练习 1 设计登录界面的表单，参考步骤如下。

- (1) 选中【项目管理器】窗口中【文档】选项卡的【表单】选项，按下【新建】按钮，在随后打开的对话框中单击【新建表单】按钮，弹出表单设计器和一个表单，其对象名为 Form1。在表单中添加一个标签对象，采用系统默认名称 Label1，并添加一个命令按钮对象，设置 Caption 属性值为“进入系统”，如图 6.49 所示。



图 6.49 “封面” 表单设计界面

(2) 接下来设置对象属性和事件代码。Form1 的属性设置如表 6.11 所示。

表 6.11 Form1 的属性设置

| 属性名          | 设置值                | 说明       |
|--------------|--------------------|----------|
| Left         | 0                  |          |
| Top          | 0                  |          |
| Heigh        | 742                |          |
| Width        | 1030               |          |
| BorderStyle  | 0—无边框              |          |
| Picture      | d:\学生成绩系统\图片\7.jpg | 设置窗口背景   |
| TitleBar     | 0—关闭               | 只显示背景和标签 |
| Window State | 2—最大化              |          |

事件代码设置如下：

● Activate 事件代码：

```
Thisform.Label1.Top=Thisform.Height/3           && 调整标签在启动窗口中的位置
Thisform.Label1.Left=(Thisform.Width-Thisform.Label1.Width)/2
Thisform.Label1.Visible=.T.
```

● RightClick 事件代码：

```
Thisform.Release           && 关闭启动窗口
```

(3) Label1 的属性设置如表 6.12 所示。

(4) 命令按钮的属性设置及事件代码如表 6.13 所示。

命令按钮的 Click 事件代码如下：

```
DO FORM d:\学生成绩系统\FORM\密码.scx
Thisform.Release
```

表 6.12 Label1 的属性设置

| 属性名       | 设置值      | 说明           |
|-----------|----------|--------------|
| Top       | 48       |              |
| Left      | 36       |              |
| Height    | 60       |              |
| Width     | 552      |              |
| Caption   | 学生成绩管理系统 |              |
| FontName  | 隶书       |              |
| FontSize  | 52       |              |
| ForeColor | 0,0,160  |              |
| BackStyle | 2—透明     | 以免遮住窗口背景     |
| Visible   | .F.      | 以免在调整标签位置时闪动 |

表 6.13 命令按钮的属性设置

| 属性名       | 设置值        |
|-----------|------------|
| Top       | 372        |
| Left      | 456        |
| Height    | 48         |
| Width     | 156        |
| Name      | Command1   |
| Caption   | 进入系统       |
| FontName  | 宋体         |
| FontSize  | 26         |
| ForeColor | 186,179,27 |
| Visible   | .T.（默认值）   |

- (5) 当完成了表单控件的添加设置与属性、事件代码的设置和编写后，关闭【表单设计器】窗口，并把表单保存在“封面.scx”文件中。在【项目管理器】窗口的【文档】选项卡下的表单中选择“封面”。单击运行按钮，就可浏览刚刚创建的表单文件的运行结果，如图 6.50 所示。



图 6.50 “封面”表单浏览界面



练习 2 设计“学生专业”表单界面，参考步骤如下。

- (1) “学生专业”表单用于输入、修改、删除、查询学生信息，窗口设计见图 6.51。在窗口上单击鼠标右键弹出快捷菜单。选择【数据环境】命令打开【数据环境设计器】窗口，在数据环境中添加“学生表”、“专业表”两个表，并建立关联，如图 6.52 所示。



图 6.51 “学生专业”表单的设计界面

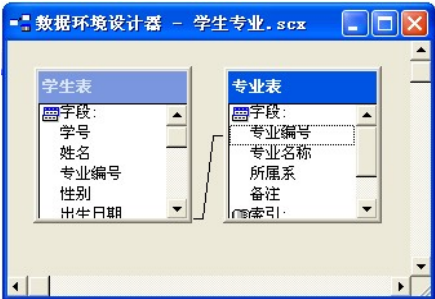


图 6.52 “学生专业”表单的数据环境设置

- (2) 设置表单属性和事件代码。StuForm 的属性设置如表 6.14 所示。

表 6.14 StuForm 的属性设置

| 属性名         | 设置值     | 说明         |
|-------------|---------|------------|
| Name        | StuForm |            |
| Caption     | 学生专业情况  |            |
| AutoCenter  | .T.     | 窗口自动居中     |
| BorderStyle | 2—固定对话框 | 无须用户调整窗口大小 |
| MaxButton   | .F.     | 同上         |

从数据环境将下列字段拖到表单中规定的位置，产生相应的标签和文本框：“学生表”中的“学号”、“姓名”、“性别”、“出生日期”、“入学成绩”、“照片”等字段；“专业表”中的“专业编号”、“专业名称”、“所属系”等字段。从数据环境产生标签和文本框，不仅速度快，而且标签的 Caption 和 Name 属性、文本框的 Name 属性都会自动设定为与源字段有关的名称，文本框也会自动与源表中的源字段绑定。

- (3) 设置“专业编号”组合框及事件代码。从【表单控件】工具栏中选择组合框控件，在窗口上的相应位置单击鼠左键，产生一个组合框 Combo1，设置其属性如表 6.15 所示。

表 6.15 组合框控件的属性设置

| 属性名           | 设置值      | 说明         |
|---------------|----------|------------|
| Name          | Combo1   |            |
| ControlSource | 学生表.专业编号 | 数据存储的字段    |
| RowSource     | 专业表.专业编号 | 组合框中的数据源字段 |
| RowSourceType | 6—字段     | 数据来源类型为字段  |
| Style         | 2—下拉列表框  |            |

Valid 事件的代码如下：

```
Thisform.Refresh
```

- (4) 设置“团员”复选框。从数据环境中将“学生.团员否”字段拖到窗口上相应的位置，会自动产生团员复选框。并且将它与“学生.团员否”字段绑定。
- (5) 设置“性别”选项按钮组。从【表单控件】工具栏中选择选项按钮组控件，在窗口相应的位置单击鼠标左键，创建选项按钮组控件，在控件上单击鼠标右键弹出快捷菜单，选择【生成器】选项，打开【选项组生成器】窗口，如图 6.53 所示，按步骤完成如下操作：第一步，将标题改为“男”、“女”；第二步，将布局改为水平排列；第三步，将值与“学生.性别”字段绑定。单击【确定】按钮完成设置。
- (6) 设置命令按钮组与事件代码。从【表单控件】工具栏中选择命令按钮组，在窗口相应的位置单击鼠标左键，创建命令按钮组控件 CommandGroup1。在控件上单击鼠标右键弹出快捷菜单，选择【生成器】选项，打开【命令组生成器】窗口。按要求第一步设置按钮数目及每个按钮的标题，如图 6.54 所示；第二步将布局设置为水平排列。单击【确定】按钮即可生成命令按钮组。

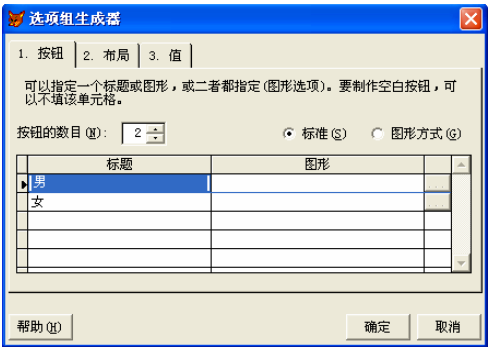


图 6.53 设置“性别”选项组

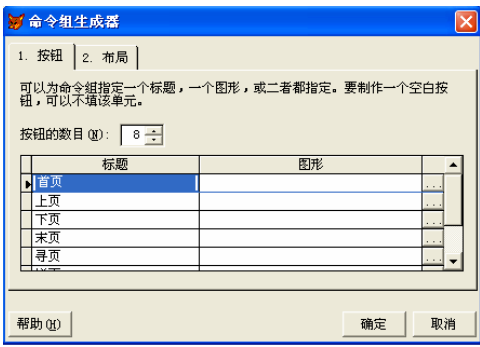


图 6.54 设置命令按钮组控件

CommandGroup1 的 Click 事件代码如下：

```
DO CASE
CASE This.Value=1      && 到首页
GO TOP
CASE This.Value=2      && 到上页
IF NOT BOF()
SKIP -1
ENDIF
CASE This.Value=3      && 到下页
SKIP
IF EOF()
SKIP
ENDIF
CASE This.Value=4      && 到末页
GO BOTTOM
CASE This.Value=5      && 按学号查询
xh=" "
DO FORM form\输入学号    && 打开学号输入窗口（见下文），关闭时将输入的学号返回
dqjlh=RECNO()
IF LEN(xh)<>0
```

```
LOCATE FOR 学号=xh
IF NOT FOUND()
    WAIT WINDOW "无此学号！"
    GO dqjlh
ENDIF
ENDIF
CASE This.Value=6      && 新增学生名单
    zy=messagebox('需要增加学生名单吗？',4+32+256,'确认')
    IF zy=6      && 对话框含“是”“否”按钮，按下“是”按钮返回 6
        APPEND BLANK
    ENDIF
CASE This.Value=7      && 删除当前学生名单
    sy=MESSAGEBOX('需要删除当前的学生名单吗？',4+32+256,'确认')
    IF sy=6
        DELETE
        PACK
    ENDIF
CASE This.Value=8      && 关闭当前窗口
    Thisform.Release
ENDCASE
Thisform.Refresh
```

练习 3 完成“输入教师编号”表单，参考步骤如下。

- (1) 单击【文件】→【新建】命令，选择【表单】项，单击【新建文件】按钮。
- (2) 在表单中添加 1 个标签、1 个文本框和 2 个命令按钮。
- (3) 设置控件的属性如表 6.16 所示。

表 6.16 各控件的属性

| 控件     | 属性           | 属性值           |
|--------|--------------|---------------|
| 文本框 1  | Name         | Text1         |
|        | PasswordChar | *             |
| 标签 1   | Caption      | 请输入需要查询的教师编号： |
|        | Name         | Label1        |
| 命令按钮 1 | Caption      | 确定            |
|        | Name         | Command1      |
| 命令按钮 2 | Caption      | 取消            |
|        | Name         | Command2      |

- (4) 双击该命令按钮，输入其 Click 事件过程的代码：

```
DO CASE
CASE This.Value=1
    jsbh=TRIM(Thisform.Text1.Value)
CASE This.Value=2
    jsbh=''
ENDCASE
```



(5) 在表单上增加另一个命令按钮，Caption 设为“退出”。其 Click 事件过程的代码如下：

```
Thisform.Release
```

(6) 运行该表单，结果如图 6.55 所示。

练习 4 如图 6.56 所示，完成“输入学号”表单。

练习 5 如图 6.57 所示，完成“按学生查询”表单。

图 6.55 “输入教师编号”表单

图 6.56 “输入学号”表单

练习 6 如图 6.58 所示，完成“选课情况”表单。

| 学号        | 姓名  | 性别 | 课程名称    | 考试成绩 |
|-----------|-----|----|---------|------|
| 080301001 | 张跃林 | 男  | VFP数据库  | 76   |
| 080301020 | 张跃林 | 男  | 多媒体计    | 87   |
| 080301000 | 张跃林 | 男  | VB.NET程 | 96   |
|           |     |    |         |      |
|           |     |    |         |      |
|           |     |    |         |      |
|           |     |    |         |      |
|           |     |    |         |      |
|           |     |    |         |      |

At the bottom is a '关闭' (Close) button.

图 6.57 “按学生查询”表单

图 6.58 “选课情况”表单

练习 7 如图 6.59 所示，完成“专业”表单。

| 专业编号 | 专业名称  | 所属系 | 备注   |
|------|-------|-----|------|
| 03   | 03    | 基础  | Memo |
| 04   | 物理    | 基础  | Memo |
| 05   | 动力    | 电力  | memo |
| 41   | 自动化   | 电力  | memo |
| 42   | 机械    | 机械  | memo |
| 01   | 计算机应用 | 计算机 | memo |
|      |       |     |      |
|      |       |     |      |
|      |       |     |      |

图 6.59 “专业”表单

## 思考与练习 6

### 一、思考题

1. 利用表单向导创建表单的基本步骤是什么?
2. 怎样在表单设计器中为表单或表单集创建新属性和新方法?
3. 如何利用表单设计器在表单中或各种容器中添加或删除各种控件?
4. 简述对象的属性、方法和事件的基本概念。
5. 简述基类、子类和父类的概念，以及类的类型。

### 二、选择题

1. 下列关于属性、方法和事件的叙述中，\_\_\_\_是错误的。
  - A. 属性用于描述对象的状态，方法用于表示对象的行为
  - B. 基于同一个类产生的两个对象可以分别设置自己的属性值
  - C. 事件代码也可以像方法一样被显式调用
  - D. 在新建一个表单时，可以添加新的属性、方法和事件
2. 假定一个表单里有一个文本框 `Text1` 和一个命令按钮组 `CommandGroup1`，命令按钮组是一个容器对象，其中包含 `Command1` 和 `Command2` 两个命令按钮，如果要在 `Command1` 命令按钮的某个方法中访问文本框的 `Value` 属性值，下面的语句\_\_\_\_是正确的。
  - A. `This.Thisform.Text1.Value`
  - B. `This.Parent.Parent.Text1.Value`
  - C. `Parent.Parent.Text1.Value`
  - D. `This.Parent.Text1.Value`
3. 下面关于数据环境和数据环境中两个表之间关系的陈述中，\_\_\_\_是正确的。
  - A. 数据环境是对象，关系不是对象
  - B. 数据环境不是对象，关系是对象
  - C. 数据环境是对象，关系是数据环境中的对象
  - D. 数据环境和关系都不是对象
4. 在表单设计器环境下，要选定表单中某选项组里的某个选项按钮，可以\_\_\_\_。
  - A. 单击选项按钮
  - B. 双击选项按钮
  - C. 先单击选项组，并选择【编辑】命令，然后再单击选项按钮
  - D. B 和 C 都可以
5. `DbClick` 事件是\_\_\_\_时触发的基本事件。
  - A. 当创建对象
  - B. 当从内存中释放对象
  - C. 当表单或表单集装入内存
  - D. 当用户双击对象
6. 在下列对象中，不属于控件类的为\_\_\_\_。
  - A. 文本框
  - B. 组合框
  - C. 表格
  - D. 命令按钮
7. 在表单 `MyForm` 的一个控件的事件或方法中，将该表单的背景色改为红色的正确命令是\_\_\_\_。
  - A. `MyForm.BackColor = RGB(255, 0, 0)`
  - B. `This.Parent.BackColor = RGB(0, 255, 0)`
  - C. `ThisForm.BackColor = RGB(255, 0, 0)`
  - D. `This.BackColor = RGB(0, 255, 0)`

8. Visible 属性的作用是\_\_\_\_\_。
- A. 设置对象是否可用
- B. 设置对象是否可视
- C. 设置对象是否可改变大小
- D. 设置对象是否可移动
9. 在引用对象时，下面\_\_\_\_\_的格式是正确的。
- A. Text1.Value = "北京"
- B. Thisform.Text1.Value = "北京"
- C. Text.Value = "北京"
- D. Thisform.Text.Value = "北京"
10. 下列控件中属于容器控件的是\_\_\_\_\_。
- A. 文本框
- B. 复选框
- C. 命令按钮
- D. 页框

### 三、填空题

1. \_\_\_\_\_是面向对象程序设计中程序运行的最基本实体。
2. 若要实现表单中的控件与某一数据表中的字段的绑定，则在设计时应先在\_\_\_\_\_设置表单的数据源为该数据表。
3. Caption 是对象的\_\_\_\_\_属性。
4. 在表单运行时，要求单击某一对象时释放表单，应\_\_\_\_\_。
5. 在对象的引用中，Thisform 表示\_\_\_\_\_。
6. 一对多表单中的表格显示的是\_\_\_\_\_的数据。
7. 数据源通常是数据库中的表，也可以是自由表、视图或\_\_\_\_\_。
8. 默认情况下，在通过表单向导设计的表单中，出现的定位按钮有\_\_\_\_\_个移动记录指针的按钮。
9. 【表单】菜单在\_\_\_\_\_时出现在 Visual FoxPro 主菜单中。
10. 使用表单设计器设计表单时，如果要对表单添加控件，应打开\_\_\_\_\_工具栏。

## 第 7 章 菜单与工具栏

菜单式界面能为用户提供友好、操作简单方便的工作环境。而对于 Visual FoxPro 来说，当操作比较简单时，一般通过控件来实现；而当完成较复杂的操作时，使用菜单具有十分明显的优势。因此，菜单的设计就显得至关重要。

### 7.1 菜单的建立

#### 7.1.1 菜单概述

##### 1. 菜单的基本组成

Visual FoxPro 支持条形菜单和弹出式菜单两种类型的菜单。一个菜单系统无论大小都由以下几部分组成。

- ① 菜单标题：即菜单的名称，放置在菜单栏中。每个菜单标题表示一个菜单。
- ② 菜单栏：位于窗口标题栏下的包含多个菜单标题的一水平条形区域。
- ③ 菜单：由一系列的选项组成，包括命令、过程或子菜单等。
- ④ 菜单项：包含于菜单之中，可以是命令、过程或子菜单等。
- ⑤ 子菜单：选择菜单项时出现的下拉菜单，由一系列菜单项组成。

##### 2. 菜单中的其他概念

一个菜单系统除了以上介绍的基本组成部分之外，还应包括以下几个部分。

① 菜单项的状态：菜单项有两种状态，当菜单项为黑色时是可用状态，当菜单项为灰色时是不可用状态。

② 访问键：每一个菜单项后面都有一个用括号括起来的英文字母，该字母代表可访问菜单项的访问键（热键），它可以是 A~Z 的任意一个英文字母。使用访问键访问某一菜单项时，按住 Alt 键，再键入括号中的英文字母即可执行相应的操作。

③ 快捷键：在某些菜单项的右侧有“Ctrl+字母”说明，这是该菜单项的快捷键标志。使用快捷键访问某一菜单项时，按住 Ctrl 键，再单击相应的英文字母即可访问。

④ 子菜单标志：在有些菜单项的右侧有一个黑色三角形，它表示该菜单项是一个子菜单，当鼠标指向该菜单项时，它将自动弹出一个子菜单。

⑤ 菜单项分隔线：在菜单中为了将某些功能相关的菜单项分在一起，在中间用一条直线和其他菜单项分隔开来，便于用户阅读使用。

##### 3. 系统菜单

典型的系统菜单一般是一个下拉菜单，由一个条形菜单和一组弹出式菜单组成。其中条

形菜单作为主菜单，弹出式菜单作为子菜单。当选择一个条形菜单选项时，激活相应的弹出式菜单。表 7.1 和表 7.2 给出了主菜单和“编辑”菜单的常见选项。

表 7.1 主菜单（\_MSYSMENU）的常见选项

| 选项名称 | 内部名称       |
|------|------------|
| 文件   | _MSM_FILE  |
| 编辑   | _MSM_EDIT  |
| 显示   | _MSM_VIEW  |
| 工具   | _MSM_TOOLS |
| 程序   | _MSM_PROG  |
| 窗口   | _MSM_WINDO |
| 帮助   | _MSM_SYSTM |

表 7.2 “编辑”菜单（\_MEDIT）的常见选项

| 选项名称  | 内部名称       |
|-------|------------|
| 撤销    | _MED_UNDO  |
| 重做    | _MED_REDO  |
| 剪切    | _MED_CUT   |
| 复制    | _MED_COPY  |
| 粘贴    | _MED_PASTE |
| 清除    | _MED_CLEAR |
| 全部选定  | _MED_SLCTA |
| 查找... | _MED_FIND  |
| 替换... | _MED_REPL  |

通过 SET SYSMENU 命令可以允许或禁止在程序执行时访问系统菜单，也可以重新配置系统菜单。

4. 菜单设计步骤

设计菜单的基本步骤一般细化为以下几点。

- (1) 打开菜单设计器。
- (2) 填写菜单栏上的菜单项。
- (3) 设置菜单项动作。
- (4) 设置菜单项特性（热键、快捷键、备注信息等）。
- (5) 设计每一个下拉菜单。
- (6) 填写下拉菜单中的菜单项。
- (7) 设置下拉菜单的菜单项动作。
- (8) 设置下拉菜单的菜单项特性（热键、快捷键、备注信息等）。
- (9) 保存菜单，生成菜单文件 (\*.mnx，\*.mnt)。
- (10) 生成菜单源程序文件 (\*.mpr)。
- (11) 运行菜单程序。

5. 菜单设计器简介

1) 【菜单名称】栏

【菜单设计器】窗口如图 7.1 所示，其中的【菜单名称】栏用于指定菜单系统中菜单项的名称。

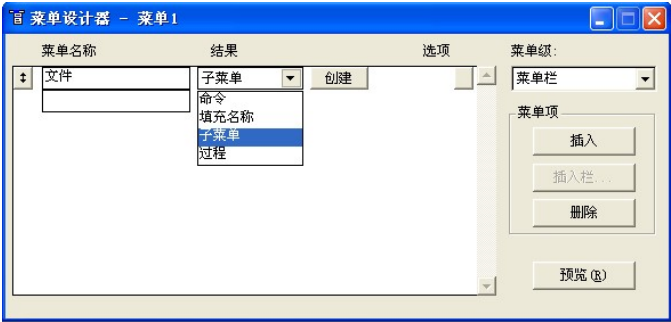


图 7.1 【菜单设计器】窗口

2) 【结果】栏

【结果】栏指定菜单项所具有的功能。

① 子菜单：用户所定义的当前菜单有子菜单即选择此项。其右侧会出现一个【创建】按钮，单击该按钮，系统会弹出子菜单设计窗口。

② 命令：当在列表框中选择了“命令”选项后，会在其右侧出现一个文本框，可在其中输入要执行的命令。即为菜单项或子菜单指定一条 Visual FoxPro 的命令，用于完成指定的操作。

③ 过程：如果所要执行的动作需要由多条命令完成，而又无相应的程序可用，此时选择“过程”最为合适，它是一组命令的集合。当在【结果】栏的下拉列表中选择了“过程”选项后，会在其右侧出现一个【创建】按钮，单击该按钮就会弹出编辑窗口，可在此窗口中输入过程代码。

3) 【选项】栏

单击【选项】栏中的按钮，会弹出【提示选项】对话框，用于设置键盘快捷键或用户定义的菜单系统中各菜单项的属性。

4) 【菜单级】组合框

【菜单级】组合框用于显示当前所处的菜单级别，从其下拉列表中选择要处理的任一级菜单。

5) 【插入】按钮

单击【插入】按钮，将在当前菜单项的前面插入一个空白的菜单项。

6) 【插入栏】按钮

单击【插入栏】按钮，会弹出【插入系统菜单条】对话框，用于插入标准的 Visual FoxPro 菜单项。

### 7) 【删除】按钮

单击【删除】按钮，将删除当前菜单项。

### 8) 【预览】按钮

单击【预览】按钮，将显示正在创建的菜单样式。

## 7.1.2 创建快速菜单

Visual FoxPro 为用户提供了快速创建菜单的功能，可将系统菜单自动添加到【菜单设计器】窗口中，为便于用户生成菜单，书中提供了系统菜单的常用功能和标题，其中许多功能可以作为应用程序的菜单功能来使用。下面介绍如何创建快速菜单。

### 1. 打开菜单设计器

选定【项目管理器】窗口中的【其他】选项卡，并选择其中的【菜单】项。单击【项目管理器】窗口中的【新建】按钮，屏幕出现【新建菜单】对话框。单击【新建菜单】对话框中的【菜单】按钮，这时系统启动菜单设计器。

### 2. 创建快速菜单

单击系统菜单栏中的【菜单】，选择其中的【快速菜单】选项，随后在弹出的【菜单设计器】窗口中加载系统菜单，供用户编辑使用，如图 7.2 所示。



图 7.2 创建快速菜单

【菜单名称】栏列出了 Visual FoxPro 系统菜单标题，后面括号中的“\<字母”为该菜单标题的访问键，如“\<F”、“\<E”等。【结果】栏显示的都是“子菜单”，表明它是一个下拉式的菜单。【编辑】按钮表示可以对结果栏的内容进行编辑。【选项】栏的按钮表示对应的菜单标题是否已在【提示选项】对话框进行了设置。

快速生成的菜单和系统菜单相同，其中的功能项可以增加，也可以修改或删除。经过适当的编辑后，一个实用的快速菜单便生成了。所创建的菜单以.mnx 为扩展名保存到磁盘上。

例如，创建一个快速菜单，主菜单中含有“文件”、“编辑”、“窗口”和“帮助”4 个菜单标题及其所包含的菜单项和子菜单。参考上述步骤，即可在图 7.3 所示的对话框进行创建。



图 7.3 示例中创建快速菜单的对话框

7.1.3 创建菜单

用户可以根据应用程序的要求，利用菜单设计器创建菜单。在创建菜单前，必须确定菜单中应包含哪些主菜单，每个主菜单中包含哪些菜单项，以及菜单项中是否含有子菜单。下面以表 7.3 所示的菜单为例介绍如何使用菜单设计器来创建菜单。

表 7.3 菜单结构表

| 主菜单 | 子菜单                  | 功能                        |
|-----|----------------------|---------------------------|
| 文件  | 新建<br>打开<br>保存<br>关闭 | 分别打开或保存指定的文件              |
| 浏览  | 教师基本情况<br>教师代课情况     | 浏览教师情况表、教师代课表、教师课酬情况表中的记录 |
| 编辑  | 教师情况表<br>教师代课表       | 编辑修改教师情况表、教师代课表中的记录       |
| 退出  | 退出                   | 退出 Visual FoxPro 系统       |

1. 创建主菜单

- (1) 打开【项目管理器】窗口的【其他】选项卡，选定【菜单】项，单击【新建】按钮，屏幕弹出【新建菜单】对话框。
- (2) 单击【新建菜单】对话框中的【菜单】按钮，屏幕出现【菜单设计器】窗口。在【菜单名称】栏中分别输入主菜单的各个菜单标题：文件、浏览、编辑、退出，如图 7.4 所示。



图 7.4 设计主菜单的菜单标题



## 2. 创建菜单项

设计好主菜单后,接下来给主菜单的各项添加菜单项,定义所要执行的命令、过程或子菜单。这里我们给菜单标题“文件”添加菜单项:新建、打开、保存、关闭。

(1) 在【菜单设计器】窗口中选择要添加菜单项的菜单标题,如选择菜单名称“文件”,在【结果】栏中选择“子菜单”选项,并单击其右侧的【创建】按钮,这时屏幕显示一个新的【菜单设计器】窗口。

(2) 新出现的【菜单设计器】窗口是用来创建二级菜单的,即菜单项,它所对应的上级菜单可从【菜单级】组合框反映出来。这里我们给主菜单“浏览”设置两个菜单项:教师基本情况和教师代课情况,如图 7.5 所示



图 7.5 “浏览”菜单中的菜单项

通过【菜单级】组合框中的“菜单栏”选项,可返回到主菜单的【菜单设计器】窗口。按照上述操作方法,可以给其他菜单标题添加菜单项。

## 3. 定义菜单项的功能

前面已经创建了菜单项及其子菜单,而在【菜单设计器】窗口的【结果】栏中共列出了 4 个选项:命令、菜单项、子菜单、过程。前面已经分别介绍了这 4 个选项的功能,下面为该菜单设置相应的选项。

### 1) 主菜单“文件”

主菜单“文件”中的菜单项“新建”、“打开”、“保存”、“关闭”设置为“菜单项 #”,分别为 \_mfi\_new、\_mfi\_open、\_mfi\_save、\_mfi\_close,如图 7.6 所示。

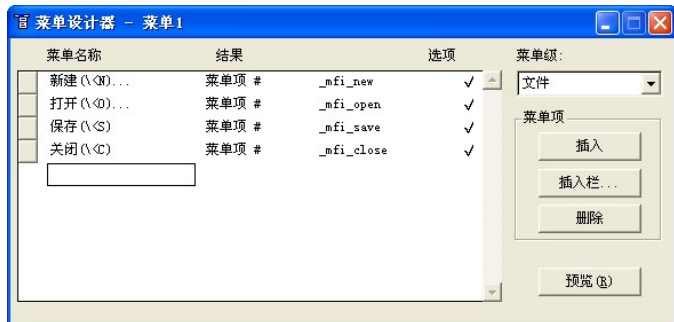


图 7.6 给菜单项指定任务

## 2) 主菜单“浏览”

主菜单“浏览”中的两个菜单项分别定义为“过程”，前面介绍了过程就是执行的动作需要多条命令完成，它是一组命令的集合。



图 7.7 过程编辑窗口

单击【结果】栏右侧的【编辑】按钮，这时屏幕出现一个过程编辑窗口。在编辑过程窗口中键入该菜单项所完成功能的命令代码。如“教师基本情况”菜单项表示打开“教师表”，并浏览记录。命令代码如图 7.7 所示。其他菜单项的含义类似，请读者自己定义相应的操作。

## 3) 主菜单“编辑”

主菜单“编辑”中的两个菜单项：“教师情况表”和“教师代课表”，分别定义为“命令”，表示执行表单，即“DO FORM d:\VFP\A\教师情况表.scx”和“DO FORM d:\VFP\A\教师代课表.scx”，如图 7.8 所示。

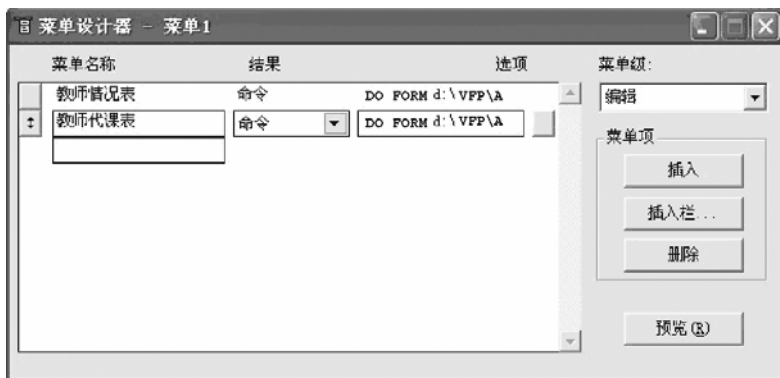


图 7.8 给菜单项指定操作命令

## 4) 主菜单“退出”

主菜单“退出”的菜单项设置为命令，所对应的命令为：SET SYSMENU TO DEFAULT。

之后，保存新创建的菜单，将文件命名为“菜单1.mnx”。保存新创建的菜单后，在【项目管理器】窗口中选择“菜单 1”并运行它，Visual FoxPro 将生成一个菜单，同时产生“菜单 1.mpr”文件。

## 4. 设置菜单的访问键

系统提供了设置访问键的功能，可以对菜单进行快速访问。建立菜单的访问键很简单，即在访问键的前面加上“<”即可。例如，要为各个主菜单标题“文件”、“编辑”、“浏览”、“退出”依次加上访问键标志，假定它们的访问键标志字母分别是：F、B、E、Q。可以按以下操作步骤设置。

(1) 在【项目管理器】窗口中选择【其他】选项卡，并选择“菜单 1”，单击【修改】按钮。

- 在【菜单设计器】窗口中选择“文件”菜单项，然后将它修改为“文件(\\<F)”。
- 在【菜单设计器】窗口中选择“编辑”菜单项，然后将它修改为“编辑(\\<B)”。
- 在【菜单设计器】窗口中选择“浏览”菜单项，然后将它修改为“浏览(\\<E)”。
- 在【菜单设计器】窗口中选择“退出”菜单项，然后将它修改为“退出(\\<Q)”。

(2) 选择系统菜单的【菜单】→【生成】命令。

(3) 在【项目管理器】窗口中选择“菜单 1”，然后单击运行按钮。此时出现了带有访问键的菜单。当在键盘上按下 Alt+Q 组合键时，就会关闭出现的消息窗口。

## 5. 设置快捷键

系统还提供了设置快捷键的功能，可以给菜单或菜单项定义快捷键。使用快捷键与使用访问键的方法类似，利用 Ctrl+字母键，可以完成快捷键的操作。快捷键与访问键的区别在于，使用快捷键可以在不显示菜单的情况下选择菜单上的某一个菜单项。例如在 Visual FoxPro 系统中按下 Ctrl+N 组合键，可建立一个新文件。下面介绍如何给菜单或菜单项定义一个快捷键。假设给菜单标题“浏览”中的“教师基本情况”和“教师代课情况”菜单项分别定义快捷键 Ctrl+G 和 Ctrl+D，可按以下步骤完成。

(1) 在【菜单设计器】窗口中单击要定义快捷键的菜单或菜单项，如选择“教师基本情况”菜单项。

(2) 单击选定“教师基本情况”菜单项右侧的【选项】按钮，屏幕显示【提示选项】对话框，如图 7.9 所示。

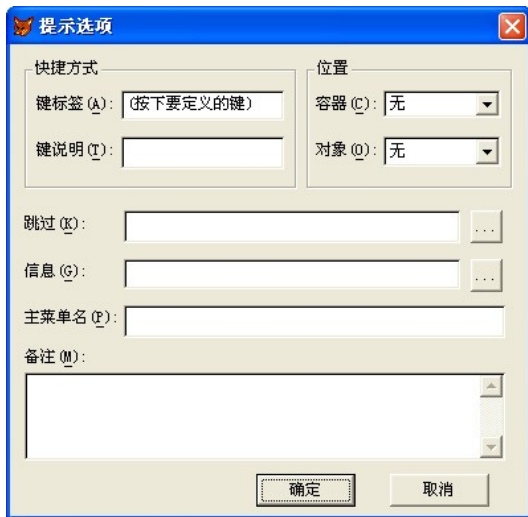


图 7.9 【提示选项】对话框

(3) 在【提示选项】对话框的【键标签】文本框中，按下一组组合键，按下的组合键就是定义的快捷键，并显示在【键标签】文本框中。如按下 Ctrl+G 组合键，【键说明】中文本框默认为 Ctrl+G，用户也可以更改键说明（如^+G）。

(4) 单击【提示选项】对话框中的【确定】按钮，返回【菜单设计器】窗口，【选项】栏中出现对号标记，表明已进行了设置。

利用同样的方法，可给“教师代课情况”菜单项定义快捷键 **Ctrl+D**。这样在不显示“浏览”菜单的情况下直接按下 **Ctrl+D** 组合键，系统就立即执行打开操作。

**提示：**不能使用 **Ctrl+J** 定义键盘快捷键，它常用于关闭某些对话框。

## 6. 设置菜单项状态

利用定义快捷键的【提示选项】对话框，还可设置菜单项的状态。在【跳过】文本框中设置一个条件表达式。在执行菜单时，将根据表达式的逻辑值来确定菜单项是否可用。当表达式值为真时，该菜单为灰色显示，表示该菜单项不可用；否则，该菜单项为黑色显示，表示该菜单项可用。另外还可以在【信息】文本框中输入状态信息，如图 7.10 所示。当用户选定该菜单项时，此信息就显示在状态栏中。

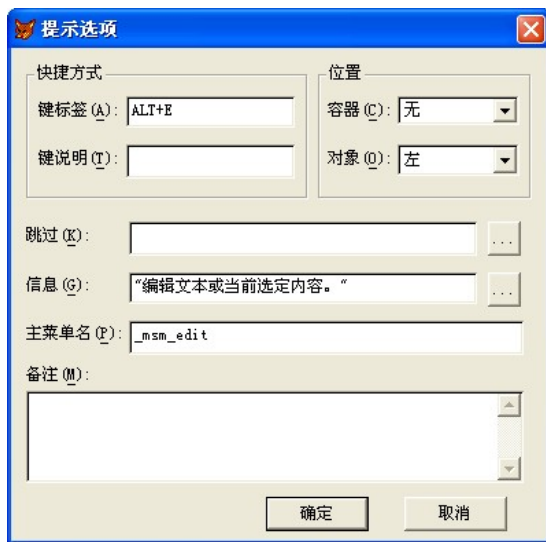


图 7.10 设置【提示选项】对话框

## 7. 对菜单项进行分组

当菜单项较多时，根据功能的需要，可以使用分隔线将功能相近的菜单划分成逻辑组。比如在菜单项“保存”上方加入一条分隔线。可按以下操作步骤进行设置。

- (1) 在【项目管理器】窗口中选择【其他】选项卡。在其中选择【菜单】项，选择要创建的菜单，单击【修改】按钮。
- (2) 在【菜单设计器】窗口中选择“文件”菜单，在其子菜单中选择“保存”菜单项然后单击【插入】按钮。
- (3) 在【菜单设计器】窗口中的【菜单名称】栏输入“\ -”。
- (4) 选择【菜单】→【生成】命令。
- (5) 单击设计器右下侧的【预览】按钮，即可查看结果。

如果对菜单进行分组，只需在需要分组的地方插入一个菜单项，将这个菜单项的菜单名称设置为“\ -”即可，如图 7.11 所示。

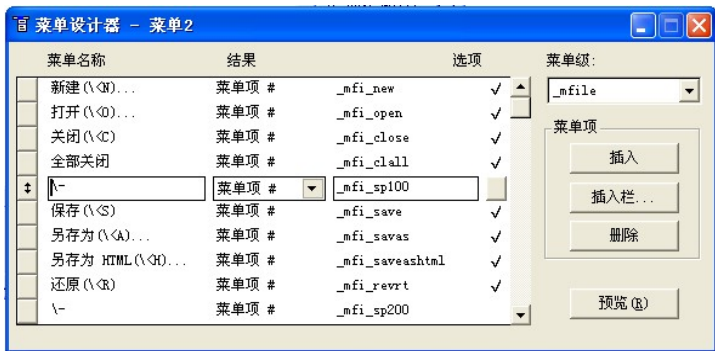


图 7.11 在菜单中显示分隔线

## 8. 添加系统菜单项

前面介绍过利用快速菜单的功能创建一个系统菜单，如果想把这个系统菜单加载到应用程序中，必须对它进行适当的修改。在利用菜单设计器创建菜单时，也可以将系统菜单中的部分菜单项加载到正在创建的菜单中，达到事半功倍的目的。比如在主菜单中添加一个主菜单标题“工具”，并设置了一个子菜单，该子菜单中包含有表、查询、表单、报表 4 个菜单项。将 Visual FoxPro 系统菜单中的这 4 项加载到【菜单设计器】窗口中，可按以下操作步骤进行设置。

(1) 在【菜单设计器】窗口中选定“工具”菜单，将其【结果】栏设置为“子菜单”，并单击其右侧的【创建】按钮，进入“工具”菜单的子菜单设计器窗口。

(2) 单击【菜单设计器】窗口中的【插入栏】按钮，屏幕显示【插入系统菜单栏】对话框。

(3) 在【插入系统菜单栏】对话框中，选定要插入的菜单项。本例中依次将表、查询、表单和报表 4 个选项插入到【菜单设计器】窗口中。利用该方法，可以比较方便地生成用户菜单项。

## 9. 定义菜单标题的位置

在【菜单设计器】窗口中，如果想改变菜单标题的位置，可以通过用鼠标拖动菜单标题前的双向箭头来调整菜单标题间的顺序。除此之外，用户在应用程序中编辑一个 OLE 对象时，要指定菜单标题的位置，可按如下步骤进行设置。

(1) 在【菜单设计器】窗口的【菜单名称】栏单击相应的菜单标题。

(2) 单击该菜单右侧【选项】栏的按钮，则出现【提示选项】对话框。

(3) 在【提示选项】对话框中有一个【位置】项，可用来指定菜单标题的位置。【位置】项中共包括 4 个单选按钮，可以从中选定一种：

- 无：指定菜单标题不设置在菜单栏上，这等同于不选择任何选项。
- 左：指定将菜单标题设置在菜单栏左边的菜单标题组中。
- 中：指定将菜单标题设置在菜单栏中间的菜单标题组中。
- 右：指定将菜单标题设置在菜单栏右边的菜单标题组中。

提示：只有在编辑 OLE 对象时，该菜单定位选项才可用。

10. 显示状态栏信息

当用户在选择菜单或菜单项时，如果在窗口下方的状态栏中显示一定的提示信息，并且该信息中包含欲选定对象的有关内容，这将给用户带来很大的方便。可按下述方法进行操作。

- (1) 在【菜单设计器】窗口的【菜单名称】栏单击相应的菜单标题。
- (2) 单击该菜单右侧【选项】栏的按钮，则出现【提示选项】对话框。有两种方法在【信息】文本框中输入要显示的提示信息：① 单击【信息】文本框右侧的浏览按钮，进入【表达式生成器】对话框，在生成器的【信息】文本框中输入提示信息；② 在【提示选项】对话框的【信息】文本框中直接输入提示信息。

提示：在文本框中输入的内容一定要放在引号"内，否则会被 Visual FoxPro 视为变量而产生错误。此外，该选项仅在【菜单设计器】窗口的【结果】栏显示“命令”、“子菜单”或“过程”时可用。

7.1.4 快捷菜单的建立

Visual FoxPro 还提供了创建快捷菜单的功能。下面介绍如何创建快捷菜单，并把它附加到指定的控件或对象中。比如我们需要创建一个包含有“剪切”、“复制”、“粘贴”和“删除”功能的快捷菜单。

1. 创建快捷菜单

- (1) 在【项目管理器】窗口的【其他】选项卡中，选择【菜单】项，并单击【新建】按钮，打开【新建菜单】对话框。单击【快捷菜单】按钮，屏幕显示【快捷菜单设计器】窗口，该窗口和【菜单设计器】窗口一样。
- (2) 在【快捷菜单设计器】窗口中，添加“剪切”、“复制”、“粘贴”和“删除”菜单项，分别指定它所完成的功能。也可以利用添加系统菜单项的方法添加 4 个菜单项，如图7.12 所示。



图 7.12 在快捷菜单中添加的菜单项

- (3) 保存新创建的快捷菜单，将文件命名为“快捷菜单 1.mnx”。
- (4) 保存新创建的快捷菜单后，在【项目管理器】窗口中选择“快捷菜单 1”并运行它，Visual FoxPro 将显示快捷菜单，同时生成“快捷菜单 1.mpr”文件。



## 2. 将快捷菜单添加到表单中

通过学习和例题的讲解，我们已经知道在创建一个菜单文件（文件扩展名为.mnx）后，如果运行该菜单，系统便会生成一个与菜单文件同名的、其扩展名为.mpr 的菜单源程序文件。利用所生成的菜单源程序文件，可以将生成的快捷菜单附加到控件或对象中。例如，我们要将快捷菜单文件“快捷菜单1.mnx”附加到前面已创建的“教师情况表”表单中，可以按以下操作步骤进行设置。

(1) 在【表单设计器】窗口中打开“教师情况表”表单文件。在【属性】窗口中，单击【方法程序】选项卡，选择 RightClick Event 属性，并双击它，这时屏幕出现过程编辑窗口。

(2) 在过程编辑窗口中输入“DO d:\学生成绩系统\MENUS\快捷菜单1.mpr”，然后保存该表单。

经过上述操作后，就可以将“快捷菜单1”文件附加到表单文件“教师情况表”表单中。运行表单文件后，利用鼠标右键单击表单中的空白区域，即可弹出此快捷菜单。执行快捷菜单中的菜单项，可以完成相应的操作，如图 7.13 所示。

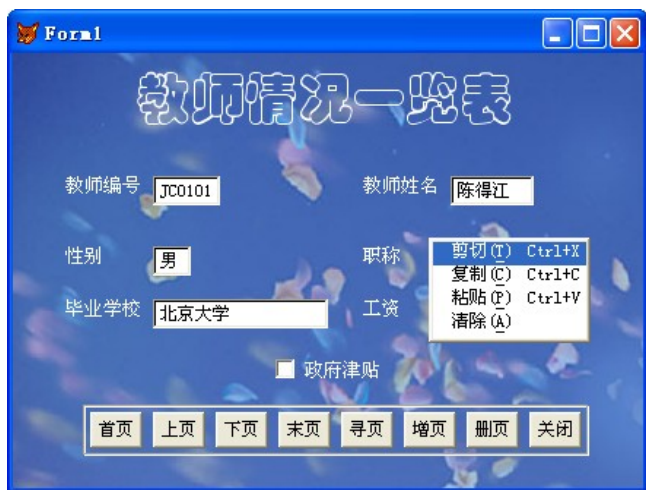


图 7.13 快捷菜单在表单中的运行结果

## 7.1.5 设置对菜单项的响应

### 1. 菜单系统的生成

用户在设计菜单时可随时单击【预览】按钮查看自己的菜单和子菜单，只是此时不能执行菜单代码。

当用户通过菜单设计器完成菜单设计后，如果用户不生成菜单源程序文件（.mpr），系统将只生成菜单文件（.mnx），而.mnx 文件是不能直接运行的。要生成菜单程序，可选择【菜单】→【生成】命令，如果用户是通过项目管理器来生成菜单，则当用户在项目管理器中选择【连编】或【运行】按钮时，系统将自动生成菜单程序。

2. 菜单系统的运行

可使用命令“DO <菜单文件名>”来运行菜单程序，菜单文件的扩展名.mpr 不能省略。

3. 为顶层表单添加菜单

为顶层表单添加下拉菜单的步骤如下。

- (1) 在菜单设计器中设计下拉菜单。
- (2) 设计菜单时，在【常规选项】对话框中选择【顶层菜单】复选框。
- (3) 将表单的 Show window 属性设置为 2，使其成为顶层表单。
- (4) 在表单的 Init 事件代码中添加调用菜单程序的命令，格式如下：

```
DO <文件名> WITH This[, "<菜单名.mpr>"]
```

例如：

```
DO d:\学生成绩系统\MENUS\主菜单.mpr WITH This, .T.
```

- (5) 在表单的 Destory 事件代码中添加清除菜单的命令，使得在关闭表单时能同时清除菜单，释放其所占用的内存空间。命令格式如下：

```
RELEASE MEMU <菜单名>[EXTENDED]
```

练习与操作

练习 1 使用菜单设计器创建一个菜单，内容如表 7.4 所示。参考步骤如下。

- (1) 单击【文件】菜单，选择【新建】→【菜单】命令，显示【菜单设计器】窗口。
- (2) 按表 7.4 要求逐行输入“菜单栏”（主菜单）的菜单名称，按需选择各个菜单的【结果】栏为“子菜单”，如图 7.14 所示。

表 7.4 主菜单结构

| 主菜单        | 菜单项   | 功能           |
|------------|-------|--------------|
| 基本信息 (\<X) | 学生名册  | 打开表单“学生基本情况” |
|            | 教师名册  | 打开表单“教师”     |
|            | 课程信息  | 打开表单“课程信息”   |
|            | \-    | 子菜单          |
|            | 关闭    | 退到表单“封面”     |
|            | 退出    | 退出 VFP       |
| 选课管理 (\<C) | 选课登记  | 打开表单“选课登记”   |
| 统计查询 (\<T) | ...   | 建立统计查询       |
| 报表打印 (\<B) | 学生登记表 | 预览报表“学生登记表”  |
|            | 学生成绩表 | 预览报表“学生成绩表”  |
| 代码维护 (\<D) | 专业代码  | 打开表单“专业”     |



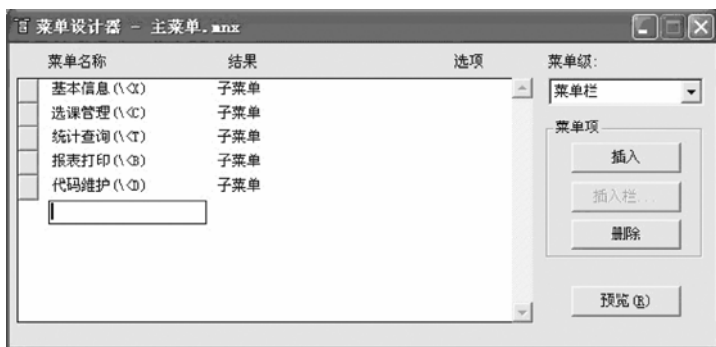


图 7.14 主菜单界面

- (3) 定义“基本信息”的子菜单，并且输入所要执行的命令或过程。单击对应“基本信息”项的【创建】按钮。如图 7.15 所示，逐行输入“基本信息”子菜单的菜单项名称，即“学生名册”等。选择各个菜单的【结果】栏为“命令”。（注意：如果命令多于一条，则应选择“过程”，从而输入若干条命令；如果还有子菜单，则应选择“子菜单”，从而定义下一级菜单。）针对“基本信息”等菜单所执行的“命令”，依次单击输入点，输入“DO FORM 学生基本情况.scx”、“DO FORM 教师.scx”、“DO FORM 课程信息.scx”、分隔线、“DO FORM 封面.scx”和“CLEAR EVENTS”。

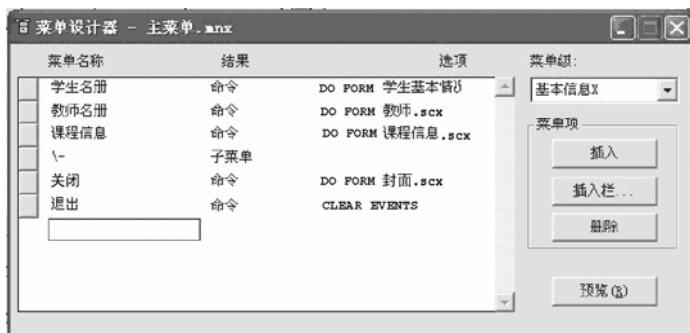


图 7.15 定义“基本信息”的子菜单

- (4) 选择“菜单栏”，回到主菜单。单击对应“选课管理”项的【创建】按钮，如图 7.16 所示。输入“选课管理”子菜单的菜单项名称，即“选课登记”。选择菜单的【结果】栏为“命令”。输入“DO FORM 选课登记.scx”。



图 7.16 定义“选课管理”的子菜单

- (5) 选择“菜单栏”，回到主菜单。单击对应“统计查询”项的【创建】按钮。如图7.17所示。输入“统计查询”子菜单的菜单项名称，即“按课程查询”、“按学生查询”、分隔线、“按课程求平均”、“按学生求平均”。选择菜单的【结果】栏为“命令”。分别输入“DO FORM 按课程查询.scx”等命令代码。



图 7.17 定义“统计查询”的子菜单

- (6) 选择“菜单栏”，回到主菜单。单击对应“报表打印”项的【创建】按钮。如图 7.18 所示，逐行输入“报表打印”子菜单的菜单项名称，即“学生登记表”、“学生成绩表”。选择各菜单的【结果】栏为“命令”。针对“学生登记表”等所要执行的“命令”，依次单击输入点，输入“REPORT FORM 学生登记表 PREVIEW”和“REPORT FORM 学生成绩表 PREVIEW”。

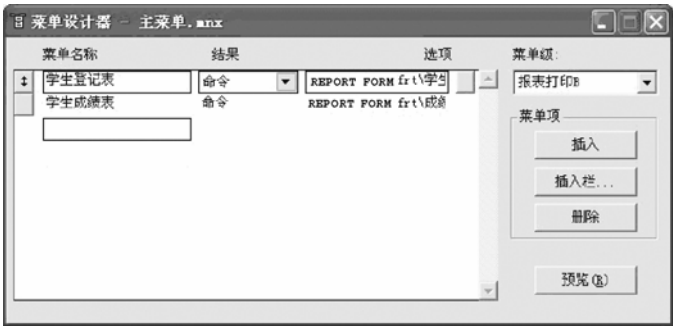


图 7.18 定义“报表打印”的子菜单

- (7) 选择“菜单栏”，回到主菜单。单击对应“代码维护”项的【创建】按钮，如图 7.19 所示。输入“代码维护”子菜单的菜单项名称，即“专业代码”。选择菜单的【结果】栏为“命令”，输入“DO FORM 专业.scx”。



图 7.19 定义“专业代码”的子菜单

练习2 为“学生成绩系统”表单建立顶层表单的菜单，参考步骤如下。

- (1) 设置“顶层表单”选项。在【菜单设计器】窗口中，单击【显示】→【常规选项】→【顶层表单】命令，使得该菜单能在顶层表单中显示。
- (2) 生成并保存所创建的菜单。在【菜单设计器】窗口中，单击【菜单】→【生成】命令，输入菜单文件名“主菜单.mnx”，单击【保存】→【生成】命令，即生成菜单源程序文件“d:\学生成绩系统\MENUS\主菜单.mpr”。
- (3) 创建“学生成绩系统”顶层表单。单击【文件】→【新建】→【表单】→【新建文件】命令，出现【表单设计器】窗口。在窗口单击鼠标右键选择【属性】命令，出现【属性】窗口，可以修改其中的几项属性：自动居中即 Autocenter=.T.，表单标题即 Caption="学生成绩系统"，表单高度即 Height=480，表单宽度即 Width=600，显示窗口即 ShowWindow="2-作为顶层表单"。双击表单中的任意位置，出现过程编辑窗口，选择初始化过程 Init，输入执行菜单程序的命令“DO d:\学生成绩系统\MENUS\主菜单.mpr WITH This, .T.”。关闭过程编辑窗口和【表单设计器】窗口，输入该表单的文件名“d:\学生成绩系统\FORM\登录.scx”，单击【保存】按钮进行保存。
- (4) 逐个打开第6章创建的“学生成绩系统”中的表单，核对和修改几项布局属性：Autocenter=.T., ShowWindow="1-在顶层表单中”。使得每个表单都能自动居中且大小相同，并且能够在顶层表单中显示。
- (5) 打开表单“密码.scx”，进入【表单设计器】窗口，双击【确定】按钮，将其 Click 过程中的命令“DO FORM d:\学生成绩系统\系统功能选择.scx”改为“DO FORM 主界面.scx”。
- (6) 运行表单及菜单。单击【程序】→【运行】命令，选择“文件类型”项为“表单”，选择第一个表单的文件名“d:\学生成绩系统\封面.scx”，单击运行按钮。随后进入“密码验证”界面、分别输入用户名“STU”和密码“123456”，单击【确定】按钮以后，显示结果如图7.20所示。



图 7.20 “学生成绩系统”的登录界面

由此可见，所有“功能”选择表单（即第一、二级功能选择表单）已被菜单取代，所有“数据”输入/输出表单（即第三、四级数据表单）依然有用。

尽管要对表单之间的连接和一些具体的处理过程进行编程，而且还要经过反复测试和修改，但只要正确一致地应用表单、菜单和顶层表单设计与操作技术，就能够按照用户需求快捷地创建用户界面十分友好的数据库应用系统。

## 7.2 工具栏

Visual FoxPro 系统提供了大量而丰富的工具栏，一些常用的命令都可以按照图形的形式出现在工具栏上。这样可以简化操作，从而提高用户的工作效率。

### 7.2.1 定制与建立工具栏

如果要创建一个工具栏，让其包含已有工具栏中没有的按钮，这就需要定制自定义工具栏来完成此任务。

#### 1. 定义工具栏类

要创建自定义工具栏，必须首先为它定义一个类。VFP 提供了一个工具栏基类，在此基础上，可以创建自己的工具栏类。创建工具栏类有以下几种方法。

- ① 在【文件】菜单中选择【新建】命令，选择“类”来创建。
- ② 单击 VFP 常用工具栏中的新建按钮进行创建；在【项目管理器】窗口中，单击【新建】按钮选择“类”进行创建。
- ③ 利用 CREATE CLASS 或 MODIFY CLASS 命令进行创建。

可以选择以上任何一种方法，之后将会弹出一个【新建类】对话框，如图7.21所示。在【类名】文本框中，输入工具栏类的名称；从【派生于】下拉列表中选择“ToolBar”；在【存储于】文本框中输入类库名。单击【确定】按钮后进入类设计器。

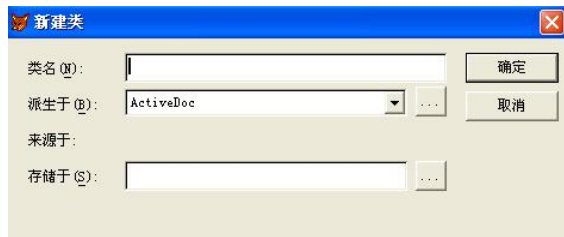


图 7.21 【新建类】对话框

#### 2. 在新建的工具栏中添加对象

(1) 在类设计器中利用【表单控件】工具栏添加对象。用鼠标单击【表单控件】工具栏中的命令按钮组，移动鼠标光标至自定义工具栏内，这时光标变为十字光标。单击鼠标左键，工具栏内出现一个带有两个命令按钮的命令按钮组，利用同样的方法也可以在工具栏内加入其他对象。

(2) 修改对象的属性。对象的属性有很多，下面列出比较重要的、常用的几个属性：

- Width 属性可以设置对象的大小。
- Picture 属性指定对象上的图标。
- DisabledPicture 属性指定按钮处于非激活状态时的图标。

- DownPicture 属性指定在按钮被按下时显示的图标。
- ToolTipText 属性指定当鼠标移到按钮上方时所显示的提示信息。

3. 设置工具栏的属性及方法

在工具栏中添加完对象以后，可以设置工具栏的属性和方法。工具栏有 28 个属性、30 个方法，下面着重说明比较常用的几个属性和方法，如表 7.5 和表 7.6 所示。

表 7.5 常用的工具栏属性

| 属性         | 说明                   |
|------------|----------------------|
| Caption    | 指定工具栏的标题             |
| ControlBox | 指定工具栏在运行时刻是否包含控制菜单框  |
| Moveable   | 指定在运行时刻用户是否可以移动工具栏   |
| Name       | 指定工具栏的名称             |
| ShowTips   | 确定是否显示工具栏中操作的提示      |
| Sizeable   | 指定在运行时用户是否可以调整工具栏的大小 |

表 7.6 常用的工具栏方法


| 方法              | 说明               |
|-----------------|------------------|
| Dock            | 可以停放或移出工具栏       |
| AfterDockEvent  | 指定在控制工具栏停放后发生的动作 |
| BeforeDockEvent | 指定在控制工具栏停放前发生的动作 |

工具栏的属性和方法还有很多，读者可以在使用中逐步体会。

4. 将工具栏加到自己的程序中

工具栏设计好以后，可将设计好的工具栏放到我们的表单集中。向表单集中添加工具栏有两种方法：一是利用表单设计器，再就是利用程序代码。

1) 利用表单设计器

首先将所建的工具栏类加到自己的项目中。打开要使用此工具栏的表单集，在【表单控件】工具栏中单击查看类按钮，在弹出的列表中单击【添加】按钮，选择工具栏类。然后再次单击查看类按钮，从显示的列表中选择该工具栏类。从【表单控件】工具栏中选择工具栏类，在表单上单击鼠标左键添加此工具栏，然后将其拖动到适当的位置。将工具栏加到表单集中以后，就可以为其按钮定义操作了，其实可以在类设计器中就为按钮定义操作，这两种方法的操作结果是一样的。

2) 利用程序代码

除了使用表单设计器之外，还可以使用程序代码在表单集中添加工具栏。如果要使用程序代码在表单集中添加工具栏，可以在表单集的 Init 事件中使用如下命令：

```
SET classlib TO 工具栏类
This ADDOBJECT("工具栏类库名", "工具栏类名")
工具栏类库名.show
```

## 7.2.2 给新工具栏命名

上述操作建立的新工具栏是由系统自动命名的。如果要为新工具栏指定一个名称，可按下列步骤进行操作。

- (1) 从【显示】菜单中选择【工具栏】命令，进入【工具栏】对话框后单击【新建】按钮。
- (2) 在【新工具栏】对话框的【工具栏名称】文本框中输入新工具栏的名称，如“表单工具栏”。单击【确定】按钮。

这样，新建的工具栏就有了自己的名称。

## 7.2.3 定义对象操作

在向工具栏类添加对象后，必须定义该对象所执行的操作。在定义操作时一般都利用【属性】窗口中 Click Event 或 DblClick Event 来设置属性。比如给“新建”按钮定义一个操作，用于打开一个数据表，可通过下列操作步骤完成。

- (1) 在【类设计器】窗口的工具栏中，选定一个定义操作的对象。这里我们选定“新建”命令按钮。在【属性】窗口的【全部】选项卡中，双击 Click Event 属性，屏幕出现一个过程编辑窗口。

- (2) 在过程编辑窗口中，输入对象所完成操作的代码。本例中的“新建”按钮所对应的代码为“CREATE”，结果如图 7.22 所示。

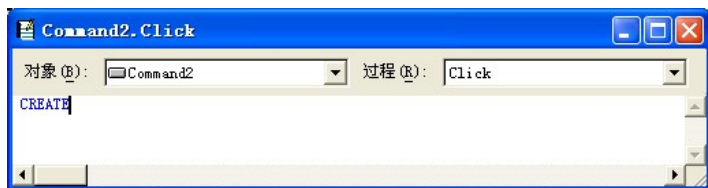


图 7.22 为“新建”按钮编写命令代码

## 练习与操作

练习 1 规划菜单系统，确定菜单系统的构成，实现步骤如下。

- (1) 启动菜单设计器，单击【文件】→【新建】→【菜单】命令，输入菜单名称（\>字母）。
- (2) 单击【查看】→【常规】选项，可以设置快捷键。
- (3) 选择顶层表单，可以确保将菜单正确地应用到表单中。
- (4) 设置菜单命令代码，为“文件”菜单项编辑子菜单，为“添加”菜单项输入命令 `append`，为“修改”菜单项输入命令 `browse`，为“退出”菜单项输入命令 `quit`。
- (5) 还可以设置菜单项启用或禁用的条件，如当数据表为空记录时，“删除”菜单无法使用；在【跳过】文本框中输入 `RECCOUNT()=0`，单击【预览】按钮预览菜单。
- (6) 单击【菜单】→【生成】命令，将文件命名为“菜单 1.mnx”并生成菜单程序，如图 7.23 所示。

练习2 创建自定义工具栏，实现步骤如下。

- (1) 单击【文件】→【新建】→【类】命令，输入类名“表单工具栏”，派生于“Toolbar”，存储于“toolib”，单击【确定】按钮，弹出【类设计器】窗口。
- (2) 打开【表单控件】工具栏，在自定义的工具栏上添加4个命令按钮“文件”、“添加”、“修改”和“退出”。
- (3) 编写按钮的对应代码，单击【文件】→【保存】命令，保存所设计的类。自定义工具栏如图7.24所示。



图 7.23 菜单界面





图 7.24 自定义工具栏

练习3 创建表单，实现步骤如下所示。

- (1) 单击【文件】→【新建】→【表单】命令，打开【表单设计器】窗口。
- (2) 单击系统菜单的【表单】→【创建表单集】命令。
- (3) 单击【表单设计器】窗口中的【数据环境】按钮，选择“学生成绩.dbf”。
- (4) 设置属性，将 Form1 的 ShowWindows 设为“2-顶层表单”。
- (5) 将 Form1 的 Init 事件代码设为“DO 菜单 1.mpr WITH This, .T.”。
- (6) 保存并命名为“表单 1.scx”，运行表单后，显示结果如图7.25所示。

练习4 运行表单，实现步骤如下所示。

- (1) 打开练习3创建的“表单 1.scx”，打开【表单控件】工具栏，单击查看类按钮，在弹出的列表中单击【添加】按钮。
- (2) 在打开的对话框中，选择练习2建立的名为“表单工具栏.vcx”的类库。
- (3) 单击【表单控件】工具栏上的工具栏按钮，则可以在表单上添加一个练习2中创建的工具栏，如图7.25所示。
- (4) 运行表单，测试菜单和工具栏，显示结果如图7.26所示。

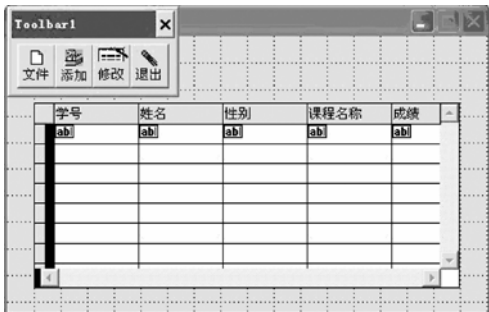


图 7.25 表单界面



图 7.26 运行结果

## 思考与练习 7

### 一、思考题

1. 在 Visual FoxPro 6.0 中, 菜单分成哪几类, 各有什么特点?
2. 快速菜单是指什么?
3. 创建快捷菜单的简单步骤是什么?
4. 怎样为菜单项“退出”设置过程代码?
5. 为顶层表单添加菜单的简单步骤是什么?

### 二、选择题

1. 在【项目管理器】窗口的\_\_\_\_\_下管理菜单。  
A. 【菜单】选项卡  
B. 【文档】选项卡  
C. 【其他】选项卡  
D. 【代码】选项卡
2. 如果要将一个 SDI 菜单附加到一个表单中, 则\_\_\_\_\_。  
A. 表单必须是 SDI 表单, 并在表单的 Load 事件中调用菜单程序  
B. 表单必须是 SDI 表单, 并在表单的 Init 事件中调用菜单程序  
C. 只要在表单的 Load 事件中调用菜单程序  
D. 只要在表单的 Init 事件中调用菜单程序
3. 有一个菜单文件 abc.mnx, 要运行该菜单的方法是\_\_\_\_\_。  
A. 执行命令 DO abc.mnx  
B. 执行命令 DO MENU abc.mnx  
C. 先生成菜单源程序文件 abc.mpr, 再执行命令 DO abc.mpr  
D. 先生成菜单源程序文件 abc.mpr, 再执行命令 DO MENU abc.mnx
4. 保存设计好的菜单后, 其生成的文件扩展名为\_\_\_\_\_。  
A. .scx 和 .sct  
B. .mnx 和 .mnt  
C. .frx 和 .frit  
D. .pjx 和 .pjt
5. 设计菜单要完成的最终操作是\_\_\_\_\_。  
A. 创建主菜单及子菜单  
B. 指定各菜单的任务  
C. 浏览菜单  
D. 生成菜单程序
6. “编辑”菜单中的“清除”菜单项的内部名称为\_\_\_\_\_。  
A. \_MED\_CLEAR  
B. \_MED\_CUT  
C. \_MED\_SLCTA  
D. \_MED\_FIND
7. 要创建快速菜单, 应当使用\_\_\_\_\_。  
A. 访问键  
B. 快捷键  
C. 事件  
D. 菜单设计器
8. 建立菜单的命令是\_\_\_\_\_。  
A. CREATE PROJECT  
B. CREATE MENU  
C. NEW MENU  
D. NEW PROJECT
9. 菜单项名称为“Help”, 要为该菜单项设置访问键 Alt + H, 则在菜单项名称中设置为\_\_\_\_\_。  
A. Alt+Help  
B. \<Help  
C. Alt+\<Help  
D. <Help



10. 所谓快速菜单是\_\_\_\_\_。
- A. 基于 Visual FoxPro 主菜单, 添加用户所需的菜单项
  - B. 快速菜单的运行速度较快
  - C. 可以为菜单项指定快速访问的方式
  - D. “快捷菜单”的另一种说法

### 三、填空题

1. 允许或禁止在应用程序执行时访问系统菜单的命令是\_\_\_\_\_。
2. 控件上的快捷菜单一般用鼠标右键单击激活, 相应的事件名称是\_\_\_\_\_。
3. 有连续的两个菜单项, 名称分别为“保存”和“删除”, 要用分隔线将这两个菜单项分组。实现这一功能的方法是\_\_\_\_\_。
4. 设有一菜单文件 mymenu.mpr, 运行菜单程序的命令是\_\_\_\_\_。
5. 保存利用菜单设计器设计好的菜单后, 其生成的文件扩展名为\_\_\_\_\_。
6. 设计自定义工具栏时, 可通过设置\_\_\_\_\_属性来给按钮添加位图或图标。
7. Visual FoxPro 支持两种类型的菜单, 分别为\_\_\_\_\_和\_\_\_\_\_。
8. 典型的菜单系统一般是一个下拉菜单, 下拉菜单通常由一个\_\_\_\_\_和一组\_\_\_\_\_组成。
9. 快捷菜单实际上是一个弹出式菜单。要将某个弹出式菜单作为一个对象的快捷菜单, 通常是在对象的\_\_\_\_\_事件代码中添加调用该弹出式菜单程序的命令。
10. 要为顶层表单设计下拉菜单, 首先需要在打开菜单设计器的状态下, 在\_\_\_\_\_对话框中选择【顶层表单】复选框; 其次要将表单的\_\_\_\_\_属性值设置为 2, 使其成为顶层表单; 最后需要在表单的\_\_\_\_\_事件代码中设置调用菜单程序的命令。

# 第 8 章 报表与标签

报表功能为用户在打印的文档中显示数据提供了灵活的途径。报表是 Visual FoxPro 数据库管理系统中常用的功能之一，是应用程序设计中的重要环节，用户可以根据需要来设计数据输出格式，形成报表文件。标签是一种特殊类型的报表，它主要用来设计各种物品标签、邮政标签等。

## 8.1 报表的设计

报表和标签由数据源和布局两个基本部分组成。数据源通常是数据表、查询文件、视图文件和临时表等。报表和标签中的记录既可以是数据表的全部记录，也可以是数据表的部分记录；既可以是数据表中的全部字段，也可以是数据表的部分字段。报表布局用来定义报表和标签的打印格式。

报表布局文件可以指定要用的数据字段、要打印的文本及页面中信息的放置方法。报表布局文件不存储每个数据字段的值，而只是存储报表中信息的布局和格式。每次运行报表时，根据报表布局文件所使用的数据源中各个字段的变化，数据字段的值也会相应地发生改变。报表文件的扩展名是.frx。

### 8.1.1 报表类型

建立报表之前，首先必须确定报表类型。报表可能是一个电话号码簿，也可能是发票这样的复杂清单。还可以建立特殊种类的报表，如标签便是一种特殊的报表。常见的报表布局类型及有关说明参见表 8.1。

表 8.1 报表常规布局类型

| 报表类型  | 说明                           | 示例                      |
|-------|------------------------------|-------------------------|
| 列报表   | 每行一个记录，每个记录的字段在页面上按水平方向放置    | 分组/总计报表、财政报表、存货清单或销售总结等 |
| 行报表   | 一行一个字段，每个记录的字段在一侧竖直放置        | 列表                      |
| 一对多报表 | 一个记录或一对多关系                   | 发票和账目                   |
| 多栏报表  | 多栏式记录，每个记录的字段沿左边缘竖直放置        | 电话号码簿                   |
| 标签    | 多列记录，每个记录的字段沿左边缘竖直放置，打印在特殊纸上 | 邮件标签                    |

### 8.1.2 利用报表向导创建报表

使用向导创建报表比较简单，用户只要按照向导提供的步骤，回答向导提出的问题，就能正确地建立报表。在建立报表的过程中，如果对前面的设计不满意，可以返回上一步进行

修改，直到满意为止。Visual FoxPro 为用户提供了两种类型的报表向导：报表向导和一对多报表向导。下面以“教师表”为例，介绍如何使用报表向导创建报表。

## 1. 创建单表报表

创建单表报表就是使用向导中的报表向导，快速地创建基于一个表或视图的报表。例如，使用报表向导创建一个“教师表”的简单报表。具体步骤如下。

(1) 打开【项目管理器】窗口，单击【文档】选项卡，选择【报表】项，并单击【新建】按钮，这时屏幕出现一个【新建报表】对话框，如图 8.1 所示。

(2) 单击【新建报表】对话框中的【报表向导】按钮，屏幕显示【向导选取】对话框，如图 8.2 所示。



图 8.1 【新建报表】对话框

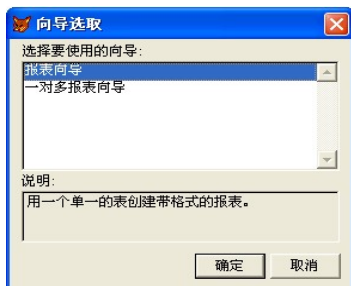


图 8.2 【向导选取】对话框

(3) 选择【向导选取】对话框中的【报表向导】选项，单击【确定】按钮，出现【报表向导】对话框，如图 8.3 所示。首先要求用户确定报表中所需的字段。这里选择“教师表”中的所有字段，单击【下一步】按钮，进入报表向导的步骤 2—分组记录，如图 8.4 所示。



图 8.3 步骤 1—字段选取

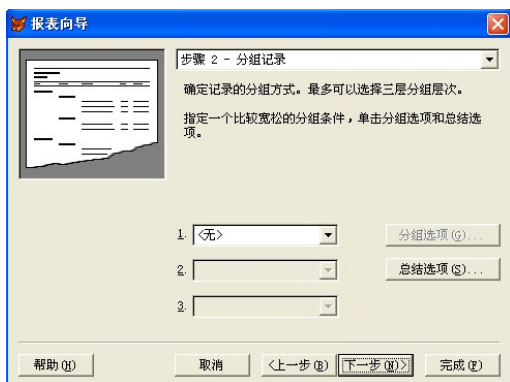


图 8.4 步骤 2—分组记录

(4) 分组记录。报表中的记录可以按一定的条件进行分组，向导提供了三个条件，这三个条件不是并列关系，而是分层关系。分组时，先按第一个条件进行分组，再将多个组中的记录按第二个条件进行分组，依次类推。单击【分组选项】按钮，可以确定分组字段的字段间隔。单击【总结选项】按钮，可以对数值字段进行求和、求平均值，以及确定报表中是否包含有小计和总计等。单击【下一步】按钮，进入报表向导的步骤 3—选择报表样式，如图 8.5 所示。

(5) 报表样式。在选择报表样式的对话框中，提供了几种报表样式。通过左上角的放大镜可以查看选取的样式是否满意，本例选择【账务式】。单击【下一步】按钮，进入报表向导的步骤4—定义报表布局，如图8.6所示。



图 8.5 步骤3—选择报表样式

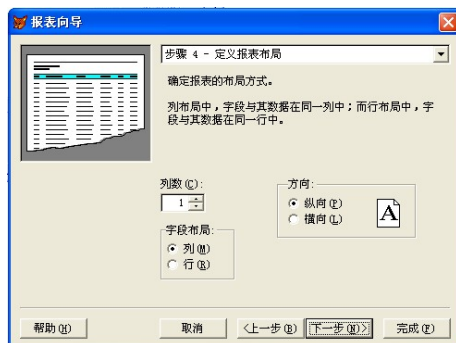


图 8.6 步骤4—定义报表布局

(6) 报表布局。在定义报表布局的对话框中，确定报表的布局。向导提供了两种布局方式：列布局和行布局。本例选择列布局、方向为纵向。并选择“列数”为1。单击【下一步】按钮，进入报表向导的步骤5—排序记录，如图8.7所示。

(7) 排序记录。在排序记录的对话框中，用于确定报表中记录的输出次序，最多设定三个用于排序的字段，按【选定字段】列表框中字段的先后顺序进行排序，排在前面的优先排序。这里按“教师编号”升序排序，单击【下一步】按钮，进入报表向导的步骤6—完成，如图8.8所示。

(8) 完成。在完成对话框中，要求用户为所创建的报表输入一个标题，该标题出现在报表的顶部，并选择适合的方式保存报表。在完成报表前，最好先单击【预览】按钮，观察报表结果，如果不满意，可单击【上一步】按钮进行修改。



图 8.7 步骤5—排序记录

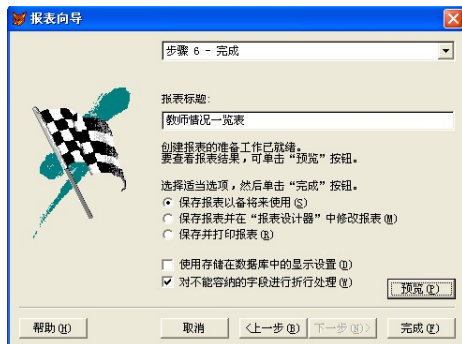


图 8.8 步骤6—完成

在【另存为】对话框中，以“教师情况表.frx”为文件名保存新创建的报表。选定该报表，单击【预览】按钮，就可以观察到报表格式及数据，如图8.9所示。

**提示：**① 利用报表向导虽然可以设计出所需要的报表，但其显示方式并不美观，所以一般情况下都要在报表设计器中进行进一步修改。

② 如果要创建基于多个表或视图的报表，必须先创建一个视图，视图中包含所需要的字段，然后再创建报表。

报表设计器 - 教师情况表.frx - 页面 1

**教师情况一览表** 01/21/09

| 教师编号   | 教师姓名 | 性别 | 职称   | 工资       | 政府津贴 |
|--------|------|----|------|----------|------|
| CC0101 | 赵晓兰  | 女  | 二级教授 | 3,334.00 | N    |
| CC0202 | 李月敏  | 女  | 讲师   | 1,678.00 | N    |
| CC0303 | 王克强  | 男  | 讲师   | 1,564.00 | N    |
| CC0404 | 刘雷燕  | 女  | 副教授  | 2,345.00 | N    |
| CS0101 | 林星明  | 男  | 教授   | 3,254.00 | N    |
| CS0202 | 赵华林  | 男  | 副教授  | 2,786.00 | N    |
| CS0303 | 李严冬  | 男  | 教授   | 3,786.00 | Y    |
| JC0101 | 陈得江  | 男  | 教授   | 3,310.10 | N    |
| JC0202 | 赵惠兰  | 女  | 副教授  | 2,587.00 | N    |
| JC0303 | 刘大林  | 男  | 讲师   | 2,000.00 | N    |
| JS0101 | 秦健中  | 男  | 副教授  | 3,120.00 | Y    |
| JS0202 | 吴谨敏  | 女  | 副教授  | 1,500.00 | N    |

图 8.9 预览报表结果

## 2. 创建一对多报表

设计与使用一对多报表是将父表和子表的记录分组并产生一个新的报表，就是将建立了关系的两个数据表中的记录打印在一个报表中。该报表上半部分的内容来自父表，下半部分的内容来自子表，两部分数据之间通过一对多关系相连接。

例如，以“学生表”为父表，以“学生选课表”为子表，创建一个一对多报表。具体步骤如下。

(1) 在【项目管理器】窗口的【文档】选项卡中，选择【报表】项，单击【确定】按钮。随后在屏幕出现的【新建报表】对话框中，单击【报表向导】按钮，在出现的【向导选项】对话框中选择【一对多报表向导】。单击【确定】按钮，进入一对多报表向导的步骤 1-从父表选择字段。

(2) 从父表选择字段。选择“学生表”为父表，并选择其中的“学号”、“姓名”和“专业编号”字段为父表字段，如图 8.10 所示。单击【下一步】按钮，进入一对多报表向导的步骤 2-从子表选择字段。

(3) 从子表选择字段。选择“学生选课表”为子表，并选定其中的“学号”、“课程编号”、“开课时间”和“成绩”字段为子表字段，如图 8.11 所示。单击【下一步】按钮，进入一对多报表向导的步骤 3-为表建立关系。

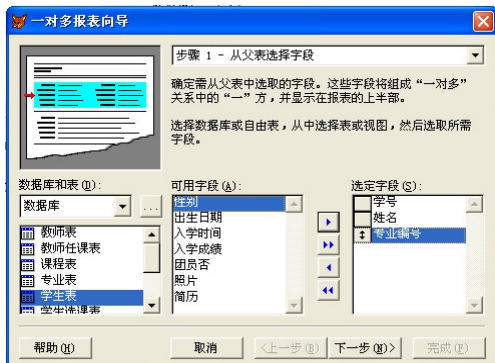


图 8.10 步骤 1-从父表选择字段

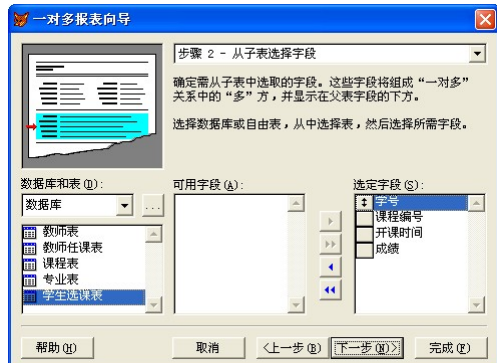


图 8.11 步骤 2-从子表选择字段

(4) 为表建立关系。为两表建立两表之间的关联表达式。这里建立的关联表达式为“学生表.学号 = 学生选课表.学号”，如图8.12 所示。单击【下一步】按钮，进入一对多报表向导的步骤4—排序记录。

(5) 排序记录。排序记录是确定父表中记录的输出次序。本例按“学生表.学号”降序排序，如图 8.13 所示。单击【下一步】按钮，进入一对多报表向导的步骤5—选择报表样式。



图 8.12 步骤3—为表建立关系



图 8.13 步骤4—排序记录

(6) 选择报表样式。用来确定报表样式及总结选项。选择“账务式”报表样式，如图 8.14 所示。单击【下一步】按钮，进入一对多报表向导的步骤6—完成。

(7) 完成。要求用户输入报表标题及保存报表方式等。本例的报表标题设为“学生选课情况表”，如图 8.15 所示。在单击【完成】按钮之前，可以先单击【预览】按钮，观察报表样式及内容是否满足要求。如果对报表结果不满意，可以单击【上一步】按钮，修改相关内容，直到满意为止。单击【完成】按钮，输入报表文件名“学生选课表.frx”，保存以上创建的一对多报表文件，运行结果如图 8.16 所示。



图 8.14 步骤5—选择报表样式



图 8.15 步骤6—完成

从报表结果可以看出，上半部分内容来自父表“学生表”，下半部分内容来自子表“学生选课表”，两个表之间通过字段建立起关联。使用报表向导所创建的报表，可以通过后面将要学到的报表设计器进行修改。



| 学生选课情况表                        |      |          |    |
|--------------------------------|------|----------|----|
| 01/21/09                       |      |          |    |
| 学号: 070401032 姓名: 刘慧琴 专业编号: 04 |      |          |    |
| 学号                             | 课程编号 | 开课时间     | 成绩 |
| 070401032                      | CC02 | 02/25/09 | 92 |
| 070401032                      | CS01 | 02/23/08 | 91 |
| 学号: 070403128 姓名: 赵晓飞 专业编号: 04 |      |          |    |
| 学号                             | 课程编号 | 开课时间     | 成绩 |
| 070403128                      | CC01 | 02/23/08 | 86 |

图 8.16 预览报表结果

### 8.1.3 使用设计器创建报表

利用报表向导创建报表比较方便、快捷，常用于简单的报表。而报表设计器可以把字段和控件添加到空白报表中，设计出美观大方、实用、复杂的报表，同时还可以对使用向导创建的报表进行修改和完善。

#### 1. 报表设计器

在 Visual FoxPro 中提供了报表设计器，允许用户通过直观的操作来直接设计报表或修改报表。Visual FoxPro 虽然提供了创建快速报表的功能，但它不能像报表向导那样单独使用，在启动快速报表之前，也必须先打开报表设计器。打开报表设计器主要有以下方法。

① 打开【项目管理器】窗口，单击【文档】选项卡，选择【报表】项，并单击【新建】按钮。在屏幕上出现的【新建报表】对话框中，单击【新建报表】按钮，这时屏幕出现【报表设计器】窗口。

② 选择系统菜单的【文件】→【新建】命令，在显示的【新建】对话框中单击【报表】项，然后单击【新建文件】按钮，这时屏幕出现【报表设计器】窗口。

#### 2. 报表设计器的基本组成

首次启动报表设计器时，报表布局中默认有三个带区：页标头、细节和页注脚，如图 8.17 所示。

- 页标头：在每一页报表的上方，常用来放置字段名标题和日期等信息。
- 细节：报表的内容。
- 页注脚：在每一页报表的下方，常用来放置页码和日期等信息。

每个带区的大小是可以改变的，将鼠标指向带区分隔条，此时鼠标指针变成垂直双箭头，拖动鼠标就可以改变带区的大小。改变大小后的带区，反映在报表上，页标头、页注脚区域和记录的行间距也随之发生改变。

#### 3. 创建快速报表

顾名思义，快速报表就是能够根据用户要求产生一个报表文件。快速报表自动地把用户

指定的字段加入到空白报表设计器中，并自动建立一个简单报表布局。这种设计报表的方法简单而又迅速。

1) 启动快速报表

单击系统菜单的【报表】→【快速报表】命令，屏幕出现【打开】对话框，用户确定创建报表所需要的数据及数据表。选择“学生选课表”，并单击【确定】按钮，屏幕出现【快速报表】对话框，如图 8.18 所示。



图 8.17 【报表设计器】窗口

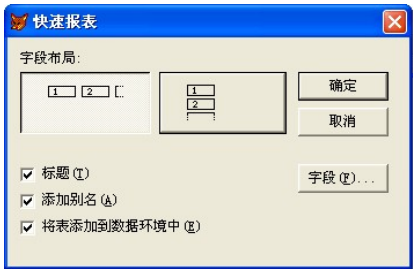


图 8.18 【快速报表】对话框

2) 设置字段布局

在【快速报表】对话框中，需要用户确定报表中字段布局方式是行布局还是列布局。行布局是字段名在字段内容的左侧，字段从上到下排列，一行一个字段，一条记录占用多行。列布局是字段名在字段内容的上方，字段从左到右排列，一列一个字段，每行一条记录。

3) 设置复选框

快速报表对话框中有三个复选框：

- 标题：是否将字段名作为页标头（列布局）或放在左侧（行布局）。
- 添加别名：是否为报表中的字段添加别名。
- 将表添加到数据环境中：是否自动将表添加到数据环境中。

4) 设置字段

通过【快速报表】对话框的【字段】按钮，可以为新创建的报表选择部分字段，单击【快速报表】对话框中的【字段】按钮，打开【字段选择器】对话框，选择报表需要的字段，如图 8.19 所示。这里选取“学生选课表”的全部字段，并选择列布局。单击快速报表对话框中的【确定】按钮，建立的快速报表显示在【报表设计器】窗口中，如图 8.20 所示。

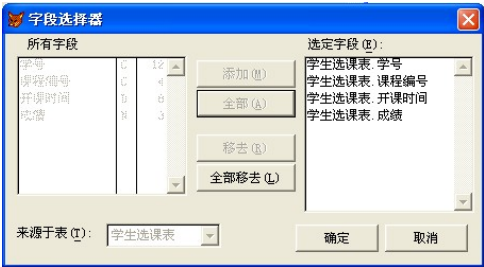


图 8.19 快速报表中的字段选取

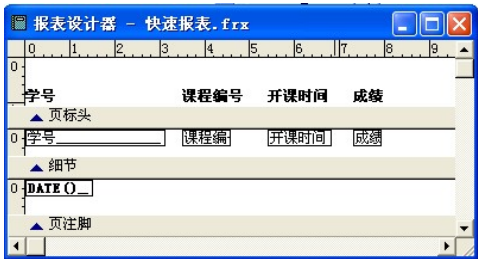


图 8.20 列布局生成的快速报表



### 5) 预览快速报表

在保存报表之前,可以通过工具栏上的打印预览按钮,预览由快速报表创建的报表,每页报表的页注脚区域显示报表当天的日期和页码,运行后的报表如图 8.21 所示。

使用快速报表功能可以快速生成一个简单的报表,但它只能基于一个表或视图来创建报表而无法建立复杂布局,并且通用型的字段内容无法显示。



图 8.21 展示了快速报表的浏览结果，显示了一个名为“报表设计器 - 报表1 - 页面 1”的窗口。窗口内是一个表格，列出了学号、课程编号、开课时间和成绩。

| 学号        | 课程编号 | 开课时间     | 成绩 |
|-----------|------|----------|----|
| 080301001 | CC01 | 02/20/09 | 96 |
| 080301001 | CC03 | 02/20/09 | 87 |
| 080301001 | CS01 | 02/23/08 | 76 |
| 080301020 | CC02 | 02/20/09 | 87 |
| 080301020 | CS02 | 02/20/09 | 88 |
| 084201002 | CC02 | 02/20/09 | 55 |
| 084201002 | CC03 | 02/20/09 | 89 |
| 074202123 | CS02 | 09/01/09 | 90 |
| 070403128 | CC01 | 02/23/08 | 86 |
| 084104112 | CC02 | 02/23/08 | 50 |
| 084104112 | CS02 | 02/20/09 | 67 |

图 8.21 快速报表浏览结果

## 8.1.4 对报表进行细节设计

使用报表设计器创建报表,用户不仅从空白报表开始设计出图文并茂、美观大方的报表,还可以在报表向导和快速报表设计的基础上,对已有的报表进行修改和完善。下面以“学生表”为例,介绍如何创建一个“学生基本情况一览表”的报表。具体步骤如下。

### 1. 启动报表设计器

前面使用快速报表功能创建报表时,已经介绍了如何启动报表设计器,这里不再赘述。

### 2. 设置数据环境

设置数据环境就是选取报表所需的数据表、视图和关系,为报表添加需要的控件。单击系统菜单的【显示】→【数据环境】命令,启动数据环境设计器。在【数据环境设计器】窗口中,单击鼠标右键,在弹出的快捷菜单中选择【添加】命令,添加报表所需要的数据源。

在打开的【添加表或视图】对话框中,选择数据表和视图,如图 8.22 所示。单击【添加表或视图】对话框中的【添加】按钮,把“学生表”添加到【数据环境设计器】窗口中,如图 8.23 所示。如果需要还可以添加多个数据表或视图。

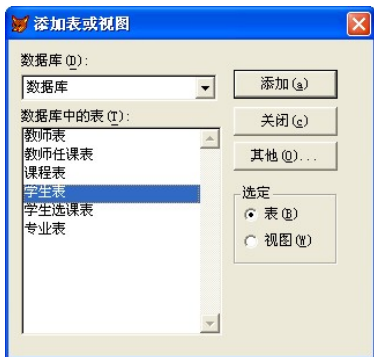


图 8.22 【添加表或视图】对话框

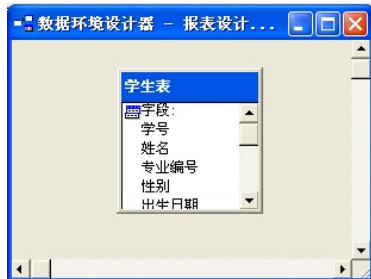


图 8.23 将“学生表”添加到数据环境设计器中

### 3. 设置报表设计器界面

添加标题和总结带区。单击系统菜单的【报表】→【标题/总结】命令,在弹出的【标题

/总结】对话框中，设置报表标题或总结。这里选择“标题带区”和“总结带区”，单击【确定】按钮。这样就在【报表设计器】窗口上添加了标题和总结两个带区。

在进行报表页面设置时，如果报表被设置成多栏报表，则在【报表设计器】窗口中还会出现两个带区：

- 列标头：用于打印在每一列（一栏）的开头。
- 列注脚：用于打印在每一列（一栏）的结尾。

4. 添加控件

报表是由各种控件组成，标题、图标、页标头、日期及时间等，都需要用添加控件的方法来实现。如果屏幕上没有显示【报表控件】工具栏，可以在【报表设计器】窗口中，单击系统菜单的【显示】→【报表控件工具栏】命令，在该选项前将出现标记“√”，屏幕上则显示【报表控件】工具栏。【报表控件】工具栏中各控件的功能如表 8.2 所示。

表 8.2 【报表控件】工具栏中各控件的功能

| 序号 | 表单控件按钮名称    | 快捷按钮图标                                                                              | 窗体控件按钮作用            |
|----|-------------|-------------------------------------------------------------------------------------|---------------------|
| 1  | 选定对象        |    | 移动或改变控件大小           |
| 2  | 标签          |    | 创建一个标签控件            |
| 3  | 域控件         |    | 创建一个文本框控件，用于单行文本    |
| 4  | 线条          |    | 在报表上画各种线条           |
| 5  | 矩形          |   | 在报表上画矩形             |
| 6  | 圆角矩形        |  | 在报表上画圆角矩形           |
| 7  | 图片/OLE 绑定控件 |  | 添加图片或随记录变化的 OLE 对象  |
| 8  | 按钮锁定        |  | 在添加多个同类型控件时，不需要多次选择 |

下面介绍如何添加各种报表控件，添加报表控件的方法类似于添加表单控件的方法。

1) 标签

在报表中，标签控件是最常用的一种控件，它可以单独使用，也可以和其他控件结合使用。在【报表设计器】窗口中，单击【报表控件】工具栏中的标签按钮，然后将鼠标指针指向标题带区，并单击鼠标左键，在光标处输入标签文本内容：“学生基本情况一览表”。利用同样的方法，在页标头带区输入“学号”、“姓名”、“专业编号”、“性别”、“入学成绩”等。

2) 域控件

报表设计中的域控件用于显示表字段、变量和表达式的内容。添加域控件有两种方法。

① 从数据环境中添加。在【数据环境设计器】窗口中，选择要添加的数据表中的字段，按住鼠标左键，将该字段拖到报表区域。本例中将数据环境“学生表”中的“学号”、“姓名”、“专业编号”、“性别”和“入学成绩”字段分别拖到细节带区内，并与页标头带区内相应的标头对齐，结果如图 8.24 所示。

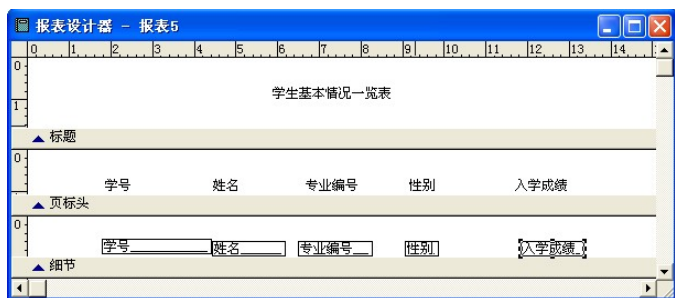


图 8.24 在【报表设计器】窗口中添加控件

② 从【报表控件】工具栏中添加。按下【报表控件】工具栏中的域控件按钮，将鼠标指针指向要放置域控件的位置，并单击鼠标左键，这时屏幕会弹出【报表表达式】对话框，如图8.25所示。在【报表表达式】对话框的【表达式】文本框中，可以直接输入一个字段表达式“学生表.学号”，也可以单击【表达式】文本框后面的浏览按钮，打开【表达式生成器】对话框。在【表达式生成器】对话框中，双击选定的字段名，如果是表达式则输入相应的表达式。如本例中选择相应的表达式为“学生表.学号”、“学生表.姓名”、“学生表.专业编号”、“学生表.性别”、“学生表.入学成绩”，如图8.26所示。依次单击【确定】按钮，返回【报表设计器】窗口。此时这几个域控件就添加在报表中，可以在程序运行时将它们的价值显示出来。

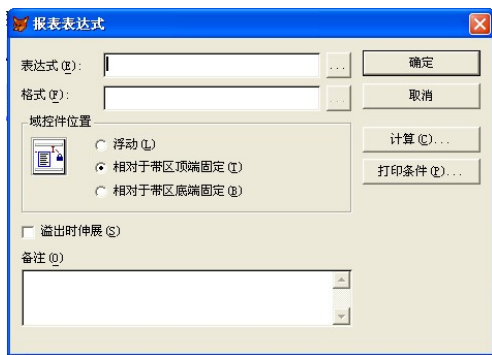


图 8.25 【报表表达式】对话框

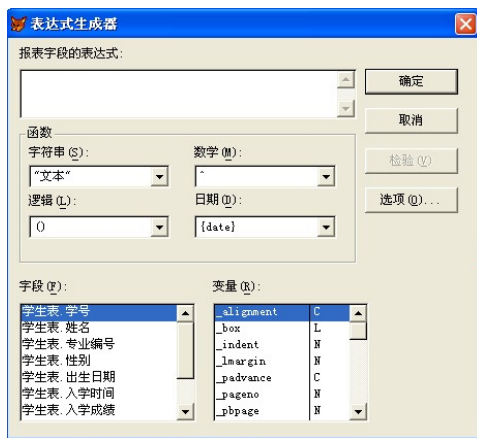


图 8.26 【表达式生成器】对话框

### 3) 图形控件

在报表中添加线条、矩形框等控件可以使报表更为清晰、美观，比如示例中的标题与页标头之间和总结中都用线条分隔开来。

按下【报表控件】工具栏中的线条按钮，分别将鼠标指针指向标题带区和总结带区，按住鼠标左键并拖动鼠标，一条直线就画出来了。利用同样的方法，可以画出矩形或圆角矩形。如果要修改线条的粗细或形状，单击系统菜单的【格式】→【绘图笔】命令，即可在其子菜单中选择适当粗细的线条或形状。

### 4) 图片/OLE绑定控件

在报表中可以添加图片、公司的标志、学校的校徽，以及随着记录的不同显示每个学生的照片等。这些会使设计的报表图文并茂，更加美观。下面介绍如何添加图片。

首先，按下【报表控件】工具栏中的图片/OLE绑定控件按钮，将鼠标指针指向标题带区的合适位置上，单击鼠标左键，屏幕出现【报表图片】对话框，如图 8.27 所示。

其次，在报表图片对话框中，指定【图片来源】区域中的“文件”，单击右侧的浏览按钮，在弹出的对话框中指定图片的文件名。可根据要求进行相应的设置。这里我们选择【缩放图片，保留形状】。

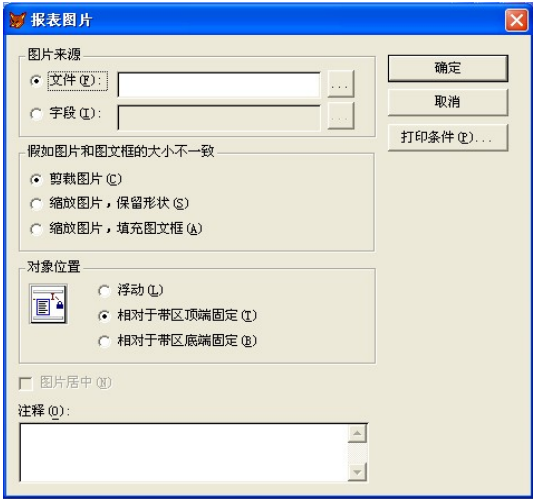


图 8.27 【报表图片】对话框

设置报表图片的各项内容后，单击【确定】按钮，则在【报表设计器】窗口中显示选定的图片，如图 8.28 所示。



图 8.28 添加图片后的【报表设计器】窗口

5. 数据分组

前面以“学生表”为基本表，使用报表向导创建了按性别分组的报表。使用报表设计器也可以根据一定的条件对记录进行分组输出，使具有相同条件的记录列在一个组中。如果对报表中的记录进行分组，那么在【报表设计器】窗口还将出现组标头、组注脚和总结三个带区。

- 组标头：放置分组的标题字段，每组打印一次。
- 组注脚：放置分组的小计，每组打印一次。
- 总结：放置各分组的总计，在报表的结尾打印一次。

设置这三个带区的方法是：单击系统菜单的【报表】→【数据分组】选项，在弹出的【数据分组】对话框中，按要求进行设置。这里我们设置分组表达式为“学生表.性别”，结果如图 8.29 所示。

## 6. 设置页面

单击系统菜单的【文件】→【页面设置】选项，屏幕显示【页面设置】对话框。其中可以设置页面的列数、每列的宽度、打印记录的顺序（指列数大于 1 时）、打印设置中的打印机驱动程序、纸张大小及方向等，如图 8.30 所示。这里在进行页面设置时，设置列数为 1 列输出。

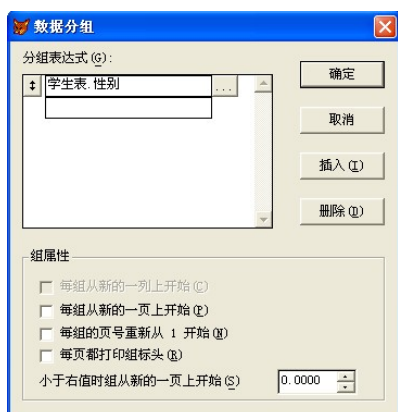


图 8.29 【数据分组】对话框



图 8.30 【页面设置】对话框

## 7. 调整带区大小

在报表设计器中，带区用来放置报表所需的各个控件。有时需要根据控件的多少、字体的大小及报表中各部分内容之间的间距来调整带区的大小。调整时，只要将鼠标指针指向要调整带区的分隔条，这时鼠标指针变成上下双箭头，按下鼠标左键并上下拖动鼠标，带区的大小随之调整。也可以双击带区分隔条，设置带区的精确高度。图 8.31 为添加控件后调整的【报表设计器】窗口。



图 8.31 添加控件后调整的【报表设计器】窗口

到此为止，利用报表设计器基本完成了该报表的设计，运行结果如图 8.32 所示。



图 8.32 使用报表设计器设计的报表

8. 设置页标头和页注脚

页标头和页注脚带区中的控件将在每个报表页中出现一次。在多页报表中，页标头和页注脚带区一般包含标签、页号和日期等。例如，假定想在页注脚带区中插入一个打印页号的域控件，方法如下。

- (1) 单击【报表控件】工具栏的域控件按钮。
- (2) 在页注脚带区中单击要插入页号的位置，弹出【报表表达式】对话框。
- (3) 单击【表达式】文本框右侧的浏览按钮，启动表达式生成器。
- (4) 在【变量】列表框中，双击系统内存变量\_pageno，使其出现在【报表字段的表达式】文本框中。
- (5) 单击【确定】按钮，返回【报表表达式】对话框。
- (6) 设置完成后单击【确定】按钮。【报表设计器】窗口的页注脚带区中将出现系统内存变量\_pageno，使用调色板工具栏将其背景色设为红色。

按照类似的方法，也可以插入一个打印当前日期的域控件，但在启动表达式生成器后，请在【日期】项中双击 DATE()函数进行设置。

提示：插入系统内存变量或日期函数后，如果想进行修改，请双击域控件，打开【报表表达式】对话框，然后进行修改。此外，系统内存变量或日期函数也可以添加到报表的其他带区中。

8.1.5 完善报表

如何进一步修饰和完善创建好的报表，是报表操作中非常重要的一个环节。当创建报表后，许多细节是不够完善的，这就需要对报表中的各个控件进行适当的修饰，达到美化报表的目的。完善报表主要包括选择、移动、删除控件，以及设置字体和字号与设置颜色、控件布局排序等。

1. 选择、移动、删除控件

在完善报表的操作中，必须遵循“先选择，后操作”的原则。选择一个控件时，将鼠标指针指向要调整的控件，并单击鼠标左键，这时在控件的周围出现控点，可以对它进行相应的



操作。也可以同时选择多个控件，对多个控件进行操作，方法是按下 **Shift** 键，然后单击每一个要选择的控件，这样依次就可以选择多个控件，将它们作为一组进行相应的操作。

选择了一个控件或一组控件后就可以进行移动操作了，按下鼠标左键并拖动鼠标，则被选中的控件移到另一个位置，如果是一组控件也同时被移动，它们的相对位置保持不变。

控件的删除操作同移动操作类似，首先选中要删除的单个控件或一组控件，然后单击键盘上的 **Del** 键，这时选中的控件一起被删除。

## 2. 设置字体和字号

在报表设计器中可以对不同栏中的文字属性进行设置。设置控件字体和字号的方法是：在【报表设计器】窗口中，选择控件，单击系统菜单的【格式】→【字体】选项，屏幕显示【字体】对话框。选择合适的文字属性进行设置，然后单击【确定】按钮。

## 3. 设置颜色

对于报表中的控件，特别是图片和标题，应该设置控件的前景和背景颜色，使设计的报表更漂亮。设置颜色的方法是首先选中要设置颜色的控件，然后单击系统菜单的【显示】→【调色板工具栏】选项，打开调色板工具栏，对控件进行相应的设置。

## 4. 布局排列

创建的报表往往需要调整各个控件的布局排列，包括控件间距、文本对齐方式等。首先选择要调整布局的一个或一组控件，然后单击系统菜单【格式】，从下拉列表中选择一种布局，然后确定对齐方式。比如我们选择“顶边对齐”，系统会使选中的一组控件以最上边的一个为参照控件，其余控件全部和它顶边对齐。

## 练习与操作

**练习 1** 利用报表设计器设计报表及报表的输出显示。参考步骤如下。

- (1) 启动报表设计器，添加标题和总结带区。单击系统菜单的【报表】→【标题/总结】选项，在弹出的【标题/总结】对话框中，设置报表标题或总结。选择【标题带区】，单击【确定】按钮，这样就在【报表设计器】窗口上添加了标题带区。
- (2) 设置数据环境。单击系统菜单的【显示】→【数据环境】选项，启动数据环境设计器。在【数据环境设计器】窗口中，单击鼠标右键，在弹出的快捷菜单中选择【添加】命令，添加报表所需要的数据源。单击【添加表和视图】对话框中的【添加】按钮，把成绩表添加到数据环境设计器中。
- (3) 调整带区大小。调整带区大小时，只要将鼠标指针指向要调整带区的分隔条上，这时鼠标指针变成上下双箭头，按下鼠标左键并上下拖动鼠标，带区的大小随之调整。也可以双击带区分隔条，设置带区的精确高度。
- (4) 设置页面。单击系统菜单的【文件】→【页面设置】选项，屏幕显示【页面设置】对话框。在对本例进行页面设置时，设定列数为 1 列输出。
- (5) 添加控件。在【报表设计器】窗口中，单击【报表控件】工具栏中的标签按钮，然后将鼠标指针指向标题带区，并单击鼠标左键，在光标处输入标签文本内容：“学生成绩一览表”；利用同样的方

法，在页标头带区输入：“学号”、“姓名”、“性别”、“课程名称”、“成绩”。按下【报表控件】工具栏中的域控件按钮，将鼠标指针指向要放置域控件的位置，并单击鼠标左键，这时屏幕会弹出【报表表达式】对话框。在【表达式生成器】对话框中，双击选定的字段名，则输入相应的表达式。

- (6) 设置页标头和页注脚。单击【报表控件】工具栏的域控件按钮，在页注脚带区中单击要插入页码的位置，弹出【报表表达式】对话框。单击【表达式】文本框右侧的浏览按钮，启动表达式生成器。在【变量】列表框中，双击系统内存变量\_pageNo，使其出现在【报表字段的表达式】文本框中。单击【确定】按钮，返回【报表表达式】对话框。设置完成后单击【确定】按钮。【报表设计器】窗口的页注脚带区中将出现系统内存变量\_pageNo，使用调色板工具栏将其背景色设为蓝色。

**练习 2** 在练习 1 的基础上设计一个报表，要求按姓名分组，计算学生的平均成绩。参考步骤如下。

- (1) 在【报表设计器】窗口的布局中添加了组标头和组注脚带区，将用于分组表示的字段添加到组标头带区内，在组注脚带区内添加域控件。添加的第一个域控件表达式为“成绩表.姓名+”的平均成绩为：“”，第二个域控件表达式为“成绩表.成绩”，并在【计算字段】对话框中设置字段为“计算平均值”。
- (2) 根据需要添加其他控件，并美化报表。
- (3) 预览报表后，如果满意，可以保存为“课程平均.frx”。运行结果如图 8.33 所示。

| 学号          | 姓名  | 性别 | 课程名称       | 成绩   |
|-------------|-----|----|------------|------|
| 080301001   | 张跃林 | 男  | VFP数据库程序设计 | 76.8 |
| 080301001   | 张跃林 | 男  | 多媒体计算机技术   | 85.6 |
| 080301001   | 张跃林 | 男  | VB.NET程序设计 | 92.8 |
| 张跃林 的平均成绩为： |     |    |            | 85.1 |
| 080301020   | 张文斌 | 男  | 数据库原理及应用   | 85.6 |
| 张文斌 的平均成绩为： |     |    |            | 85.6 |

图 8.33 报表运行结果

**练习 3** 创建基于“学生表”的一个报表文件“学生登记表.frx”，结果如图 8.34 所示。参考步骤略。

学生登记表

01/21/09

学号: 080301001      照片:

姓名: 张跃林

专业编号: 自动

性别: 男

出生日期: 10/23/89

入学时间: 09/01/08

入学成绩: 589

团员否: .T.

简历: 该学生被评为2007年度

图 8.34 “学生登记表.frx” 报表的运行结果



练习4 创建基于“学生选课表”和“课程表”的一个报表文件“成绩单.frx”。结果如图8.35所示。参考步骤略。



| 学号        | 姓名  | 性别 | 课程名称       | 考试成绩 | 平时成绩 | 总评成绩 |
|-----------|-----|----|------------|------|------|------|
| 070401032 | 刘慧琴 | 女  | 数据库原理及应用   | 92   | 80   | 89.6 |
| 070403128 | 赵晓飞 | 男  | VB.NET程序设计 | 86   | 80   | 84.8 |
| 074202123 | 李林萍 | 女  | 大学计算机应用基础  | 90   | 80   | 88.0 |
| 080301001 | 张跃林 | 男  | VFP数据库程序设计 | 76   | 80   | 76.8 |
| 080301001 | 张跃林 | 男  | 多媒体计算机技术   | 87   | 80   | 85.6 |
| 080301001 | 张跃林 | 男  | VB.NET程序设计 | 96   | 80   | 92.8 |
| 080301020 | 张文斌 | 男  | 数据库原理及应用   | 87   | 80   | 85.6 |
| 080401081 | 孟江浩 | 男  | 数据库原理及应用   | 65   | 80   | 68.0 |
| 084104112 | 郝明雅 | 女  | 大学计算机应用基础  | 50   | 80   | 56.0 |
| 084104112 | 郝明雅 | 女  | VB.NET程序设计 | 67   | 80   | 69.6 |
| 084201002 | 夏利华 | 女  | 多媒体计算机技术   | 89   | 80   | 87.2 |

图 8.35 “成绩单.frx”报表的运行结果

## 8.2 创建标签

Visual FoxPro 还可以利用标签文件作为输出格式。标签是一种特殊类型的报表，所以标签文件与报表文件在形式和设计方法上都非常相似，用户可以根据自己的需要，设计诸如邮寄标签、物品标签、个人名片、磁盘标签等。下面简要介绍利用标签向导和标签设计器创建标签。

### 8.2.1 使用标签向导创建标签

标签向导可以使用户迅速、方便地创建出美观、实用的标签。下面以“教师表”为例介绍如何使用标签向导设计标签。操作步骤如下。

(1) 在【项目管理器】窗口的【文档】选项卡中选择【标签】项，再按下【新建】按钮，这时屏幕出现【新建标签】对话框。在【新建标签】对话框中，单击【标签向导】按钮。进入标签向导的步骤1—选择表。

(2) 选择表。屏幕显示的【标签向导】对话框中要求选择数据库和数据表，这里选择数据库中的“教师表”，如图8.36所示。单击【下一步】按钮，进入标签向导的步骤2—选择标签类型。

(3) 选择标签类型。在选择标签类型的对话框中选择“英制”或“公制”，并确定标签的型号、大小和列数，单击【新建标签】按钮，还可以自定义一个标签。这里选择“英制”及4144型号的标签，如图8.37所示。单击【下一步】按钮，进入标签向导的步骤3—定义布局。

(4) 定义布局。在定义布局的对话框中，用来确定标签的布局。标签中包括文本、字段、标点符号、空格、换行等内容。假定本例标签中包含“教师编号”、“教师姓名”、“性别”、“职称”4个字段内容，标签之间用虚线间隔。为使标签美观，可以选择适当的字体，如图8.38所示。单击【下一步】按钮，进入标签向导的步骤4—排序记录。



图 8.36 步骤 1-选择表

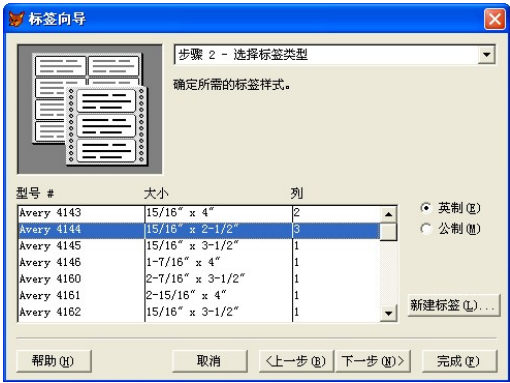


图 8.37 步骤 2-选择标签类型

(5) 排序记录。屏幕出现排序记录的对话框，确定记录的输出顺序。这里选择“教师编号”为排序字段，如图 8.39 所示。单击【下一步】按钮，进入标签向导步骤 5-完成。

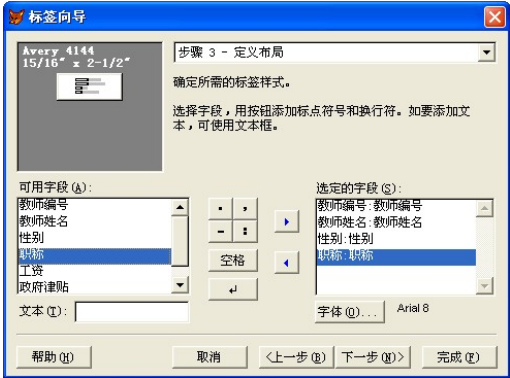


图 8.38 步骤 3-定义布局

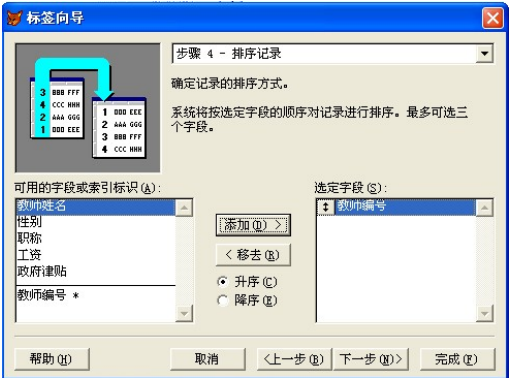


图 8.39 步骤 4-排序记录

(6) 完成。在保存标签之前，可以先预览标签内容，如果设计不理想，可返回上一步进行修改，如图 8.40 所示。

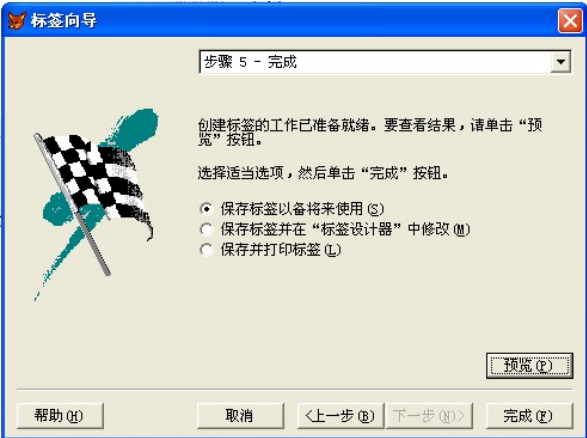


图 8.40 步骤 5-完成

(7) 保存。在【另存为】对话框中，将创建的标签保存到“教师标签.lbx”文件中，单击【保存】按钮。此时可以看到在“标签”选项前多了一个“+”号，单击“+”号，可以看到刚建立的标签文件“教师标签.lbx”。按上述操作步骤，一个简单的标签就建立好了。可以预览一下结果，如图 8.41 所示。



图 8.41 标签预览结果

## 8.2.2 利用标签设计器创建标签

标签向导可以根据用户的设定自动生成标签文件，但是由标签向导生成的标签文件形式过于简单，很难满足用户的特殊要求。而系统提供的标签设计器却可以弥补这个不足。

使用标签设计器和使用报表设计器在设计过程中有很多相似之处，在这里通过一个设计实例来说明如何利用标签设计器设计标签。

### 1. 打开标签设计器

打开标签设计器的方法主要有以下两种。

① 利用【文件】→【新建】命令启动标签设计器。在【新建】对话框中，选择【标签】项，并单击对话框右侧的【新标签】按钮，进入标签设计器。

② 在项目管理器中启动标签设计器。打开【项目管理器】窗口，单击【文档】选项卡中的【标签】项，单击对话框右侧的【新建】命令按钮。在【新标签】对话框中，单击【新标签】按钮，进入标签设计器。

### 2. 使用标签设计器

现在我们设计一组标签，每个标签以“教师表”为基础，包括“教师编号”、“教师姓名”、“性别”和“职称”4个字段。设计步骤如下。

(1) 进入【标签设计器】窗口后，首先出现【新建标签】对话框，要求用户选择标签布局，这里选择“4144”，然后单击【确定】按钮，如图 8.42 所示。如果在【标签设计器】窗口中没有所需的标签规格，可以在【新建标签】对话框中加入新的标签规格，标签的新规格将保存在相应的资源文件中。

(2) 按照以前学习过的方法启动数据环境设计器，在【数据环境设计器】窗口中单击鼠标右键选择【添加】命令，在弹出的【添加表或视图】对话框中，选择“教师表”，把“教师表”添加到【数据环境设计器】窗口中，如图 8.43 所示。



图 8.42 【新建标签】对话框

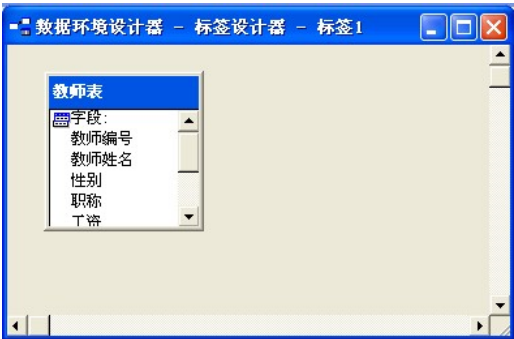


图 8.43 将“教师表”添加到【数据环境设计器】窗口

(3) 单击系统菜单的【显示】→【工具栏】选项，在【工具栏】对话框中，选择“报表控件”并单击【确定】按钮，这样【报表控件】工具栏就显示在窗口中了，如图 8.44 所示。

(4) 用鼠标单击选定字段并将其拖到【标签设计器】窗口中的适当位置。窗口背景中的网格线可以帮助确定位置。另外，在带区中设有标尺，可以更精确地定位对象的水平和垂直位置。在本例中，将“教师表”中的 4 个字段拖到【标签设计器】窗口，作为标签中出现的字段。使用鼠标单击字段周围的边框将出现几个控点，可以通过它们调节字段的位置。

(5) 利用【报表控件】工具栏中的标签工具为每个字段添加说明标签。

(6) 单击【报表控件】工具栏中的图片/OLE 绑定控件按钮，可以在报表上的相应的位置上画出图片放置区域，如图 8.45 所示。



图 8.44 【标签设计器】窗口

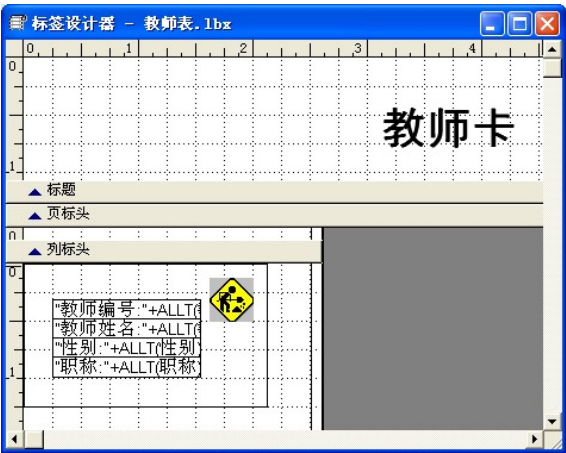


图 8.45 添加控件后的【标签设计器】窗口

(7) 单击系统菜单的【显示】→【浏览】命令，则可以浏览使用标签设计器设计的结果，如图 8.46 所示。



图 8.46 预览标签结果

## 练习与操作

练习 1 指出利用标签向导设计标签的步骤。

练习 2 指出利用标签设计器设计标签的步骤。

练习 3 回答并完成下列问题：

- (1) 报表和标签有何异同？
- (2) 标签向导步骤 3 中的“文本”文本框的作用是什么？
- (3) 将标签设计器设计的标签的标题“教师卡”放到列标头中会有什么效果？如果放到细节带区中又会怎样？

## 8.3 报表和标签的输出

报表和标签是数据的最终输出格式，它们将被定位于打印机，在前面的各节中也已经学习了有关于报表和标签的不少知识，现在来看看如何打印报表或标签。

### 8.3.1 页面设置

在打印之前，应考虑页面的外观，例如页边距、纸张类型和所需的布局等。如果改变了纸张大小和方向设置，就应考虑该文字方向是否适合于所选择的纸张大小。页面设置包括纸张的选择、版式、页边距等。

#### 1. 设置左页边距

启动 Visual FoxPro，单击【文件】→【打开】命令，在打开的对话框中选择需要打开的报表或标签文档。选择【文件】→【页面设置】命令，弹出如下图所示的【页面设置】对话框，如图 8.47 和图 8.48 所示。在此对话框中的左页边距中输入边距数值，页面布局将按新的边距显示。





图 8.47 报表的页面设置



图 8.48 标签的页面设置

## 2. 选择纸型和方向

在【页面设置】对话框中，单击【打印设置】按钮，打开【打印设置】对话框。在【大小】列表框中选择纸张大小。系统默认打印方向为纵向，如果改变了纸张方向，可在【方向】区选择“横向”，设置完毕后单击【确定】按钮。

### 8.3.2 打印输出报表和标签

要打印报表和标签文件，必须将该文件在设计器中打开，即在项目管理器中选取该文件后，单击【修改】命令按钮。

#### 1. 报表和标签的打印方式

- ① 在系统菜单中选择【文件】→【打印】命令。
- ② 在系统菜单中选择【报表】→【运行报表】命令。
- ③ 在报表或标签设计器中单击鼠标右键，在弹出的快捷菜单中选择【打印】命令。

#### 2. 报表和标签的打印步骤

- (1) 无论采用上述哪种方法，都将弹出【打印】对话框，如图 8.49 所示。
- (2) 单击对话框中的【选项】按钮，进入【打印选项】对话框，如图 8.50 所示。
- (3) 在【打印内容】区域可以设置欲打印文件的类型和文件名。使用鼠标单击其中一项，在【类型】组合框中选定该项。在【文件】文本框中可以直接输入欲打印文件的名字，也可以单击该文本框右侧的浏览按钮，进入【打开】对话框进行选择。单击【打印选项】对话框中的【选项】按钮，可以在弹出的【报表和标签打印选项】对话框中设置一定的条件，用来筛选符合条件与范围的记录数据。

- (4) 在【报表和标签打印选项】对话框中包括以下部分。

- All: 打印全部记录。
- Next: 打印当前记录往下的  $n$  个记录。 $n$  需要在右侧的微调框中设定。
- Record: 打印第  $n$  条记录。 $n$  需要在右侧的微调框中设定。

- Rest: 从当前记录开始打印到文件尾。
- For 文本框: 设定打印条件, 符合该条件的任一条记录都将被打印。
- While 文本框: 设定打印条件, 当记录符合该条件时将被打印, 直到遇到不合条件的记录便停止打印。



图 8.49 【打印】对话框

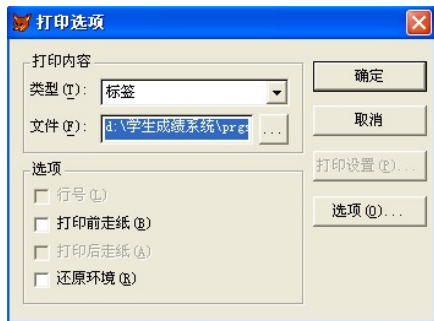


图 8.50 【打印选项】对话框

## 练习与操作

练习 1 在打印之前为什么要进行页面设置? 页面设置包括哪些内容?

练习 2 指出在报表和标签打印选项的对话框中包括哪些内容? 各是什么?

练习 3 回答并完成下列问题:

- (1) 报表和标签的打印方式是什么?
- (2) 报表和标签的打印步骤是什么?
- (3) 范围 Next 是指什么?

## 思考与练习 8

### 一、思考题

1. 报表由哪几部分组成?
2. 标签与报表有何不同之处和相同之处?
3. 简述使用报表向导、报表设计器创建报表格式文件的操作过程。
4. 什么是数据环境?
5. 怎样在报表中插入图片?

### 二、选择题

1. 为了在报表中加入一个文字说明, 应该插入一个\_\_\_\_。  
A. 表达式控件      B. 域控件      C. 标签控件      D. 文本控件
2. 报表文件的扩展名是\_\_\_\_。  
A. .rpt      B. .frx      C. .rep      D. .rpx

3. 在项目管理器的哪个选项卡下管理报表? \_\_\_\_。

- A. 【报表】选项卡      B. 【程序】选项卡      C. 【文档】选项卡      D. 【其他】选项卡

项卡

4. 为了在报表中加入一个表达式, 应该插入一个\_\_\_\_。

- A. 表达式控件      B. 域控件      C. 标签控件      D. 文本控件

5. 报表设计器中不包含在基本带区的有\_\_\_\_。

- A. 标题      B. 页标头      C. 页脚注      D. 细节

6. 报表是按照\_\_\_\_处理数据的。

- A. 数据源中记录出现的顺序      B. 主索引  
C. 人的愿望      D. 逻辑顺序

7. 利用向导可创建的报表有\_\_\_\_。

- A. 单表      B. 多表      C. 单表、多表      D. 以上都不是

8. 如果报表中的数据需要排序或分组, 应在\_\_\_\_中进行相应的设置。

- A. 报表的数据源      B. 库表      C. 视图或查询      D. 自由表

9. 预览报表的命令是\_\_\_\_。

- A. PREVIEW REPORT      B. REPORT FORM PREVIEW  
C. PRINT REPORT PREVIEW      D. REPORT PREVIEW

10. 不能作为报表数据源的是\_\_\_\_。

- A. 数据库表      B. 视图      C. 查询      D. 自由表

### 三、填空题

1. 报表定义的因素有\_\_\_\_、报表的布局。

2. 在数据分组时, 数据源应根据\_\_\_\_创建索引, 并且在报表的数据环境中进行设置(设置 Order 属性)。

3. 报表向导分为: 报表向导、\_\_\_\_。

4. 定制报表控件时, 可使用【格式】菜单中\_\_\_\_命令对控件进行字体属性的设置。

5. 对于报表中不需要的控件, 选定后按下\_\_\_\_键可删除控件。

6. 对报表进行数据分组后, 报表会自动包含的带区是\_\_\_\_。

7. 在\_\_\_\_中, 不但可以设计报表布局, 规划数据在页面上的打印位置, 而且可以添加各种控件。

8. 报表可以在打印机上输出, 也可以通过\_\_\_\_浏览。

9. 在开发应用程序时, 常用到的 OLE 技术是指\_\_\_\_技术。

10. 在报表中利用\_\_\_\_控件加入图片。



# 第 9 章 应用程序的编译与发布

怎样将已经设计好的数据库、表单、报表、菜单等组件在项目管理器中连编成一个完整的应用程序？可见，如果要开发一个比较完整的实用数据库应用系统，必须掌握基本的开发方法和步骤。

## 9.1 应用程序的开发

### 9.1.1 系统开发步骤

对于不同的系统，其开发步骤有所不同，就是不同的设计者去开发同一个系统，开发步骤也会有所不同。但一个实用的数据库应用系统应该包括以下几部分：

- 确定系统的功能和需求，设计一个或多个数据库。
- 设计用户界面。例如输入表单、显示表单、工具栏和菜单等。
- 进行事务处理。例如查询、统计和计算。
- 设计输出形式与界面。例如浏览、排序和布局报表、标签等。
- 设计主程序。设置应用程序系统环境和起始点。

下面介绍如何将上面这些组件集成起来生成一个应用程序。

#### 1. 建立应用程序目录结构

一个完整的应用程序，可能包含了多种类型的文件，如数据库文件、表文件，以及菜单、表单、报表、位图等文件。不能图省事而把所有文件都存放在一个文件夹下，而应该根据文件类型建立一个层次清晰的目录结构，以方便日后的修改和维护工作。例如可将数据库文件（.dbc）、表文件（.dbf）和索引文件（.cdx）都存储在 DATA 目录下，如图 9.1 所示。



图 9.1 应用程序的目录结构

#### 2. 利用项目管理器组织应用系统

一般情况下，完整的应用程序需要为用户提供一个菜单、一个或多个用于数据输入和输出的表单。为了保证数据的完整性和安全性，还需要为某些事件编写代码，提供特定功能。同时允许用户从数据库读取数据，可能还需要提供查询和报表输出功能。在完成所有的功能组件的设计、开发和测试之后，就可以使用项目管理器对应用程序进行集成和连编了。

使用项目管理器组织应用系统的步骤是

- (1) 创建或打开一个项目。
- (2) 将已经开发好的各个模块或部件通过项目管理器添加到该项目中。
- (3) 在项目管理器中自上而下地调试各个模块。

所谓“自上而下”是指先调试可以独立运行的模块单元，如一个输入表单、一个输出报表。然后再调试运行调用它们的模块单元，如主菜单。

## 9.1.2 连编应用程序

当应用程序中的各个模块调试成功以后，就可以对整个项目进行联合调试并编译了，这项工作 Visual FoxPro 中称为连编项目。

### 1. 设置文件的“排除”与“包含”


在 Visual FoxPro 的项目管理器中可以将添加的文件设置为“排除”或“包含”。而“排除”与“包含”是互斥的。刚添加的数据库左边都有一个符号，表示此项从项目中排除。就是说文件名左侧有符号的是“排除”，若文件名左侧没有符号，表示该文件是“包含”，如图 9.2 所示。



图 9.2 数据库名前的“排除”符号

项目连编之后，设置为“包含”的文件是只读文件，设置为“排除”的文件可以由用户修改。新添加到项目中的数据库文件的默认设置就是“排除”，表示允许用户修改，如果设置为“包含”，则相应数据库文件在程序执行期间不允许用户修改。

将标记为“排除”的文件设置成“包含”的操作方法是：在项目管理器中选定文件，单击鼠标右键，从快捷菜单上选择【包含】命令。对于没有“排除”标记的文件，可以用类似的方法从快捷菜单上选择【排除】命令来设置。

一般情况下，可执行程序，例如表单、报表、查询、菜单和程序文件应该在应用程序文件中设置为“包含”，而数据文件则设置为“排除”，当然根据需要也可以将某些数据文件设置为“包含”。所有不允许用户更新的文件应设置为“包含”。

### 2. 设置主程序

主程序是一段特殊的程序，它的任务是初始化环境、显示初始的用户界面、控制事件循环，并在退出应用程序时恢复原始的开发环境。

用户运行应用程序时，首先执行的是主程序文件，之后由主文件依次调用其他组件。因此任何一个完整的应用程序都必须有一个主程序文件，而且是唯一的。

可以定义为主文件的有程序文件、菜单、表单及查询。

设置主程序有以下两种方法。

① 在项目管理器中选中要设置的主程序文件，然后从【项目】菜单或快捷菜单中选择【设置主文件】命令。设置完毕，主文件将被自动设置为“包含”。

② 在【项目信息】对话框的【文件】选项卡中选中要设置的主程序文件，然后右击鼠

标，在弹出的快捷菜单中选择【设置主文件】命令。设置为“包含”的文件才能激活快捷菜单的【设置主文件】命令。

由于主文件是只读的，所以不能将主文件设置为“排除”。设置为主文件的文件名用黑体显示。

### 3. 连编项目

连编项目的目的是让 Visual FoxPro 系统对项目的整体性进行测试。连编以后，除了被设置为“排除”的文件，项目包括的其他文件将合成为一个应用程序文件。

连编项目的步骤如下。

- (1) 打开【项目管理器】窗口。
- (2) 选中事先设置好的主程序文件，单击【连编】按钮，弹出【连编选项】对话框，如图 9.3 所示。
- (3) 在【连编选项】对话框中，选择【重新连编项目】单选按钮。
- (4) 根据需要选择【显示错误】复选框和【重新编译全部文件】复选框。
- (5) 单击【确定】按钮完成操作。

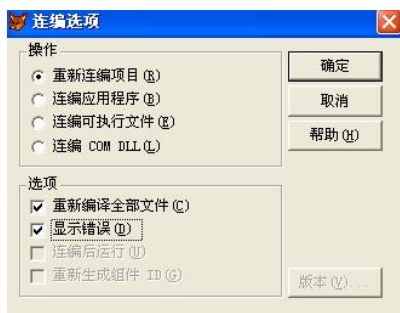


图 9.3 【连编选项】对话框

上述操作也可以通过在【命令】窗口执行“BUILD PROJECT <项目名>”来完成。

例如，可在【命令】窗口中执行“BUILD PROJECT d:\学生成绩系统\学生成绩.pjx”，从而连编项目“学生成绩.pjx”。

如果在项目连编过程中发生错误，必须纠正或排除错误，并且反复进行“重新连编项目”操作，直至连编成功。

### 4. 连编应用程序

连编应用程序可以选择两种文件形式：应用程序文件（.app）和可执行文件（.exe），前者必须在 Visual FoxPro 环境下运行，而后者既可以在 Visual FoxPro 环境下运行，也可以在 Windows 下运行。

连编应用程序的操作步骤如下。

- (1) 打开【项目管理器】窗口。
- (2) 单击【连编】按钮，弹出【连编选项】对话框。
- (3) 在【连编选项】对话框中选择【连编应用程序】或【连编可执行文件】单选按钮。
- (4) 单击【确定】按钮，并在【另存为】对话框中输入生成文件的文件名。若在第三步选择的是【连编应用程序】，则文件名以.app 为扩展名，而若在第三步选择的是【连编可执行文件】，则文件名应以.exe 为扩展名。

上述操作也可以在【命令】窗口通过执行命令完成。连编应用程序的命令是

BUILD APP <应用程序名> FROM <项目名>

连编可执行文件的命令是

```
BUILD EXE <可执行文件> FROM <项目名>
```

## 5. 运行应用程序

### 1) 运行应用程序文件 (.app)

运行 .app 文件前要先启动 Visual FoxPro，然后从【程序】菜单中选择【运行】命令，选择要执行的应用程序。也可以在【命令】窗口执行“DO <应用程序文件名>”运行应用程序。

### 2) 运行可执行文件 (.exe)

产生的 .exe 文件既可以像应用程序文件一样运行，也可以在 Windows 中执行，而不需要启动 Visual FoxPro。

运行“学生成绩系统.app”应用程序文件之后，进入主界面，如图 9.4 所示。



图 9.4 运行应用程序后显示的主界面

## 9.1.3 主程序设计

主程序是整个应用程序的入口点，主程序负责初始化环境、显示初始的用户界面、控制事件循环、组织主程序文件。当退出应用程序时，恢复原始的开发环境。

### 1. 初始化环境

主程序文件需要做的第一件事情就是对应用程序环境进行初始化。在 Visual FoxPro 中，环境的设置一般使用 SET 命令。SET 命令可以在系统环境下截取，而不需要一句句地输入。从系统环境中截取命令的方法是

① 选择系统菜单【工具】→【选项】命令。按下 Shift 键的同时，选择【选项】对话框中的【确定】按钮。此时在 Visual FoxPro 的【命令】窗口中显示了若干 SET 命令。

② 从【命令】窗口中将需要的 SET 命令复制并粘贴到主程序中。

除了设置环境之外，一般还需要进行变量初始化、建立默认的路径，以及打开需要的数据库、表及索引等操作。

## 2. 显示初始的用户界面

程序运行时显示的第一个人机交互界面就是初始的用户界面。初始的用户界面可以是个菜单，也可以是一个表单或其他用户组件。

在主程序中使用 **DO** 命令运行一个菜单，或者使用 **DO FORM** 命令运行一个表单，就可以初始化用户界面了。例如：

```
DO maim.mpr  
DO FORM start.scx
```

## 3. 控制事件循环

应用程序的数据环境建立之后，将显示出初始的用户界面，这时需要建立一个事件循环来等待用户的操作。

控制事件循环的方法是执行 **READ EVENTS** 命令，该命令使系统开始处理诸如单击鼠标、数据输入等用户事件，直到 **CLEAR EVENTS** 命令出现，循环停止。

可以将 **READ EVENTS** 作为初始过程的最后一个命令。如果在初始过程中没有 **READ EVENTS** 命令，应用程序运行后将返回到操作系统中。

在启动了事件循环之后，应用程序将处于最后显示的用户界面元素控制之下。如果在主文件中没有包含 **READ EVENTS** 或类似的命令，那么在开发环境中可以正确地运行应用程序。但是，如果要在菜单或者主屏幕中运行应用程序，程序可能显示片刻，然后退出。

## 4. 组织主程序文件

如果在应用程序中使用一个程序文件（.prg）作为主文件，必须保证该程序能够控制应用程序的执行过程。

**提示：**在启动事件循环之前建立一个方法来退出事件循环。应在界面上存在一个执行结束事件循环的 **CLEAR EVENTS** 命令机制，如一个“退出”按钮或菜单命令。该命令将挂起 Visual FoxPro 的事件处理过程，同时将控制权返回给执行该命令并开始事件循环的程序。

## 练习与操作

**练习 1** 将数据库应用系统所涉及的全部文件添加到项目管理器中。参考步骤如下。

(1) 打开已创建好的项目文件“学生成绩系统”。

(2) 在【项目管理器】窗口的【数据】选项卡中添加数据库表和查询文件。

- 选中【数据】选项卡，单击其列表中的【数据库】项。单击【添加】按钮，弹出【打开】对话框。将已创建的“学生选课库.dbc”数据库添加到项目管理器中。
- 选中【数据】选项卡，单击【数据库】项中的表，单击【添加】按钮，弹出【打开】对话框。在对话框的【搜寻】项中选择“教师表.dbf”所在的存储路径，找到并双击该文件。同样的方法将“教师选课表”、“课程表”、“学生表”、“学生选课表”、“专业表”添加到数据库中。

- 在【数据】选项卡的列表中，单击【查询】项。单击【添加】按钮，弹出【打开】对话框。在对话框的【搜寻】项中选择“学生成绩.qpr”所在的存储路径，找到并双击该文件。

(3) 在【项目管理器】窗口的【文档】选项卡中添加表单和报表文件。

- 选中【文档】选项卡，单击其列表中的【表单】项。单击【添加】按钮，将“封面.scx”文件添加到【表单】项中。使用同样的方法将“登录.scx”、“封面.scx”、“学生基本情况.scx”、“课程信息.scx”、“教师情况表.scx”、“专业.scx”、“输入教师编号.scx”、“输入课程编号.scx”等添加到项目管理器中。
- 在【文档】选项卡的列表中，单击【报表】项。单击【添加】按钮，将“成绩单.frx”文件添加到【报表】项中。使用同样的方法将“学生情况报表.frx”和“教师情况报表.frx”添加到项目管理器中。

(4) 在【项目管理器】窗口的【其他】选项卡中添加菜单文件。选中【其他】选项卡，单击其列表中的【菜单】项。单击【添加】按钮，将“主菜单.mnx”文件添加到【菜单】项中。

**练习 2** 整个应用系统以主界面表单为主操作界面（主文件），使用菜单命令的执行方式调用所有系统功能及其程序和过程。参考步骤如下。

- (1) 在【项目管理器】窗口的【代码】选项卡中，单击其列表中的【程序】项。单击【添加】按钮，将“主菜单.mpr”文件添加到【程序】项中。
- (2) 使用鼠标右键单击程序组中的“主菜单”文件名，并在弹出的快捷菜单中执行【设置主文件】命令。这样就将其设置为主文件。

**练习 3** 新建主程序，参考步骤如下。

打开已创建好的项目文件“学生选课系统”。在【项目管理器】窗口的【代码】选项卡中单击【新建】按钮，弹出【程序】对话框。编写如下主程序：

```
CLEAR ALL
SET TALK OFF
SET DEFAULT TO sys(2003)           && 设置程序的默认路径，编译时用
*SET DEFAULT TO 'd:\学生成绩系统'   && 设置程序的默认路径，调试时使用
PUBLIC xh, jsbh, kcbh
&& xh 用于存储输入的学号；jsbh 用于存储输入的教师编号；kcbh 用于存储输入的课程编号
DO FORM 封面                        && 调用封面程序
MODIFY WINDOW SCREEN TITLE "学生成绩系统" && 设置主窗口的标题
DO 主菜单.mpr                       && 调用菜单
READ EVENTS                         && 建立事件循环
*QUIT                               && 退出 Visual FoxPro
SET SYSMENU TO DEFA
```

**练习 4** 将项目管理器保存为“学生成绩.pjx”，连编生成“学生成绩系统.exe”文件后运行。参考步骤如下。

- (1) 保存项目文件“学生成绩.pjx”，并在连编生成“学生成绩系统.exe”文件后运行。
  - 单击【项目管理器】窗口中的【连编】按钮，弹出【连编选项】对话框。
  - 在对话框中选择【连编可执行文件】单项，然后单击【确定】按钮。

- 在弹出的【另存为】对话框中，选择适当的存储路径，并在【文件名】文本框中输入“学生成绩系统”，然后单击【保存】按钮保存文件。
- (2) 单击【项目设计器】窗口右上角的关闭按钮，关闭项目设计器。
- (3) 执行【程序】→【运行】命令，并在弹出的【运行】对话框中双击“学生成绩系统.exe”文件名，即可进入该数据库应用系统的主界面。

## 9.2 应用程序生成器

使用应用程序向导能够生成一个项目和一个 Visual FoxPro 应用程序框架，然后使用应用程序生成器可以添加已生成的数据库、表、表单和报表等组件。

### 9.2.1 使用应用程序向导

创建一个新项目有两种途径，一种是仅创建一个项目文件；另一种是使用应用程序向导生成了一个项目和一个 Visual FoxPro 应用程序框架。

#### 1. 使用应用程序向导创建项目和应用程序框架

启动应用程序向导的操作步骤如下。

- (1) 从【文件】菜单中选择【新建】命令。
- (2) 在【新建】对话框中选中【项目】单选按钮，单击【向导】按钮。
- (3) 在【应用程序向导】对话框中选中【创建项目目录结构】复选框，如图 9.5 所示。
- (4) 在对话框的项目名称中直接输入新项目的名称，或单击浏览按钮以后，在【选择目录】对话框中指定一个已经存在的项目文件。
- (5) 单击【应用程序向导】对话框上的【确定】按钮。



图 9.5 【应用程序向导】对话框

#### 2. 应用程序框架

应用程序框架中包含了所有必须的和可选的元素，目的是使所开发的应用程序更有效，使用起来得心应手。

【项目管理器】窗口的【类】选项卡中包含了许多类，只有通过应用程序向导建立的项目才会有这些类。当然还不止这些，还有许多其他的文件，这些文件组成了应用程序框架。

#### 3. 应用程序生成器的功能

经过上述步骤建立项目文件时，可以同时打开【项目管理器】和【应用程序生成器】窗口。此时的项目管理器包含了许多程序元素（应用程序框架）。



应用程序生成器与项目管理器一起提供了以下几项功能。

- ① 添加、编辑或删除与应用程序相关的组件，如数据库表、表单、报表等。
- ② 设定表单和报表的外观样式。
- ③ 加入常用的应用程序元素，包括启动画面、【关于】对话框、【收藏夹】菜单、【用户登录】对话框和【标准】工具栏。
- ④ 提供应用程序的作者和版本等信息。

## 9.2.2 应用程序生成器

【应用程序生成器】窗口中包括【常规】、【信息】、【数据】、【表单】、【报表】和【高级】6个选项卡。

### 1. 【常规】选项卡

如图 9.6 所示，在【常规】选项卡中可以设置的内容包括以下几点。

- 【名称】文本框：应用程序的名称。
- 【图像】文本框：图像文件名，该图像文件将显示在启动画面和关于对话框中。
- 【应用程序类型】区域：应用程序的运行方式。可供选择的运行方式有三种：
  - 正常——应用程序（.app）。
  - 模块——添加到已有的项目中，或将被其他程序调用的程序。
  - 顶层——在 Windows 桌面上运行的可执行程序（.exe）。
- 【常用对话框】区域有 4 个复选框，负责选择在应用程序中是否包括下列内容：显示屏幕，快速启动，关于对话框，用户登录。
- 【图标】按钮：指定应用程序的图标。

### 2. 【信息】选项卡

如图 9.7 所示，【信息】选项卡用于指定应用程序的产品信息，包括作者、公司、版本、版权和商标。



图 9.6 【常规】选项卡



图 9.7 【信息】选项卡



### 3. 【数据】选项卡

【数据】选项卡用于指定应用程序的数据源，并可以选择表单和报表的样式，如图 9.8 所示。

数据库向导和表向导用于创建应用程序所需的数据库和表。关闭向导后，表格中将列出新数据库中的表。

【选择】、【清除】和【生成】三个按钮分别用于选择要在应用程序中使用的已有数据库或表、删除表格中列出的表、按照指定样式生成表单或报表。

【表单样式】和【报表样式】两个下拉列表中列出了可选择的表单样式和报表样式，供用户选择。

### 4. 【表单】选项卡

【表单】选项卡用于指定菜单类型、启动表单的菜单、工具栏，以及表单是否可有多个实例。需要为每个列出的表单分别设置所需的选项，如图 9.9 所示。

【名称】文本框输入表的名称。【添加】、【编辑】、【删除】三个按钮分别用于将已有的表单添加到应用程序中、在【表单设计器】窗口中修改选定的表单及删除表单。其中有 5 个复选框：【单个实例】、【使用定位工具栏】、【使用定位菜单】、【在文件新建对话框中显示】和【在文件打开对话框中显示】，可以根据需要选择。



图 9.8 【数据】选项卡

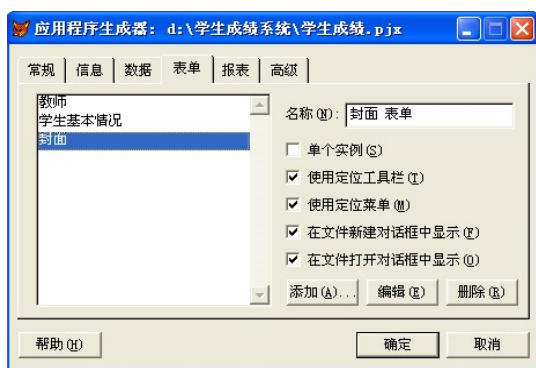


图 9.9 【表单】选项卡

### 5. 【报表】选项卡

【报表】选项卡（见图 9.10）用于指定在应用程序中使用的报表名称。其中：

- 在【名称】文本框中输入表单的名称。
- 【在打印报表对话框中显示】复选框指定选定的报表名称是否出现在应用程序的【打印报表】对话框中。
- 【添加】、【编辑】、【删除】三个按钮分别用于将已有报表添加到应用程序中、在【报表设计器】窗口中修改选定的报表和删除报表。

6. 【高级】选项卡

【高级】选项卡（见图 9.11）指定帮助文件名和应用程序的默认目录，还可指定应用程序是否包含常用工具栏和“收藏夹”菜单。其中：

- 在【帮助文件】文本框内可以指定应用程序帮助文件的名称和路径。
- 在【默认的数据目录】文本框内指定应用程序数据文件的默认目录。
- 【菜单区域】有两个复选框：【常用工具栏】和【收藏夹菜单】，前者指定应用程序是否具有常用工具栏，后者指定应用程序是否具有“收藏夹”菜单。
- 【清理】按钮：使应用程序生成器中所做的修改与当前活动项目保持一致。

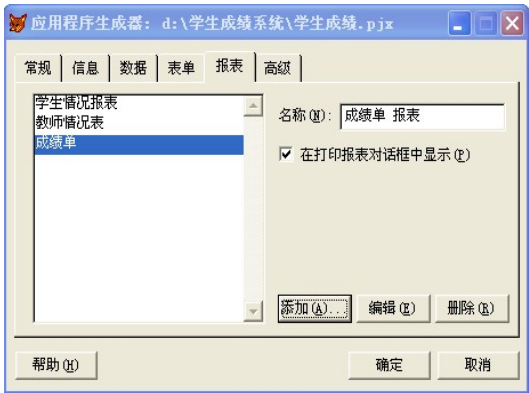


图 9.10 【报表】选项卡



图 9.11 【高级】选项卡

7. 重新启动应用程序生成器

应用程序生成器是可重入的。打开项目管理器之后，使用下面三种方法就可以再次启动应用程序生成器。

- ① 在项目上单击鼠标右键，选择快捷菜单上的【生成器】选项。
- ② 在系统菜单中选择【工具】→【向导】选项，单击【全部】选项，然后在【向导选取】对话框中选择【应用程序生成器】，如图 9.12 所示。
- ③ 按下 Alt+F2 组合键。

只有使用应用程序向导创建的项目，在启动应用程序生成器后，才会包含【常规】、【信息】、【数据】、【表单】、【报表】、【高级】6 个选项卡。如果不是使用应用程序向导创建的项目，那么在启动应用程序生成器后，只包含【数据】、【表单】、【报表】三个选项卡。



图 9.12 重新启动应用程序生成器

9.2.3 建立Visual FoxPro应用程序

我们将通过项目管理器来建立一个简单的实例，说明使用应用程序向导和程序生成器创建并修改应用程序、最后生成应用程序的一个完整的 Visual FoxPro 应用程序的建立步骤。

## 1. 建立应用程序的目录结构

即使一个很小的应用程序，也会涉及到多种类型的文件，如数据库、表及菜单、表单、报表、位图等。如果把这些文件都放在一个目录下，将会给以后的修改、维护工作带来很大的不便。因此，需要建立一个层次清晰的目录结构，让不同类型的文件各归其所。将前几章中创建的数据库文件、表文件和索引文件等移入 DATA 目录中。

## 2. 使用应用程序向导创建项目

在 Visual FoxPro 的常用工具栏中，单击新建按钮打开【新建】对话框，在该对话框的文件类型选项中选择【项目】，单击【向导】项，在弹出的【应用程序向导】对话框中，为新建的项目文件选择“Visual FoxPro”目录。输入项目名“学生成绩”，选中【创建项目目录结构】复选框，并单击【确定】按钮。系统会自动生成一个“学生成绩”项目和项目结构，如图 9.13 所示。

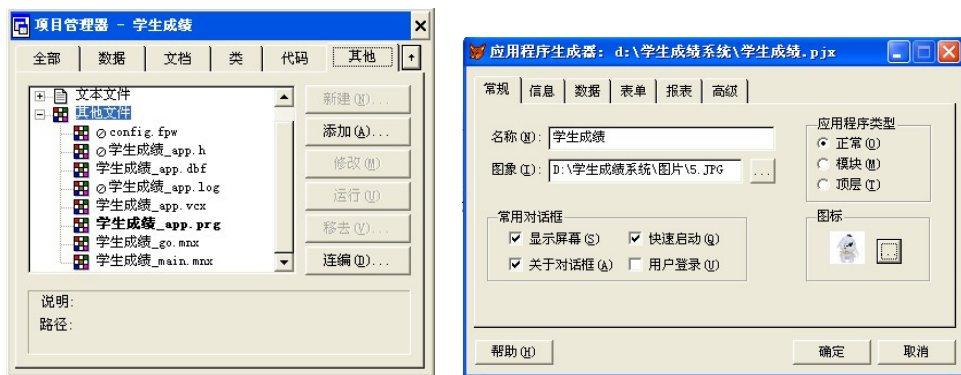


图 9.13 用程序向导创建的项目

## 3. 增加项目元素

Visual FoxPro 应用程序至少需要：一个菜单、一个表单和一个主程序。当然数据库也是应用程序的一部分。下面将把这些元素依次添加到“学生成绩”项目中。

### 1) 将数据库加入到项目中

单击【项目管理器】窗口的【数据】选项卡，并选中列表中的【数据库】项，再单击右侧的【添加】按钮，在弹出的【打开】对话框中选取先前创建的“学生成绩”数据库，确认后，列表中【数据库】项的左边出现一个加号，单击加号展开，可以看到“学生成绩”数据库已经加入到项目中，依次展开目录中的各项，即可利用右侧的按钮打开、关闭或修改数据库或修改表结构，以及进行浏览表等操作，如图 9.14 所示。

### 2) 创建表单

单击【文档】选项卡并选中【表单】项，单击【新建】按钮，在弹出的【新建表单】对话框中单击【表单向导】按钮，在弹出的【向导选取】对话框中选取【表单向导】，选中“学生成绩”数据库的“教师基本情况表”，单击单箭头按钮将“教师基本情况表”中的几个字段移动到右侧的【选定字段】列表框中。使用鼠标拖动选定字段左侧的方块可以改变字段在表

单中的显示次序。单击【下一步】按钮，进行表单样式的选择。在这里选择了“浮雕式”和“图片按钮”。接着选取“编号”作为排序字段。最后，单击【预览】按钮来预览该表单，如果不满意，可以单击【上一步】按钮返回前面的向导重新选择；否则选取“保存并运行表单”，单击【完成】按钮。在弹出的【另存为】对话框中，选择“FORM”目录，并给此表单命名为“教师基本情况表”，单击【保存】按钮后可以看到完成后的表单如图 9.15 所示。



图 9.14 添加数据库和表



图 9.15 “教师基本情况表”表单

将鼠标移动到表单底部的图形按钮上，很快就会看到此按钮的提示文本，而屏幕底部的状态栏上有更详细的按钮用途说明。

3) 添加一个菜单

在应用程序中，尤其是在 Windows 的应用程序中，菜单是必不可少的。单击“教师基本情况表”表单的退出按钮，以关闭表单并返回到【项目管理器】窗口。单击最后一个选项卡【其他】，选中【菜单】项并单击右侧的【添加】按钮。在弹出的对话框中，选择路径，选择该路径下的“menu”菜单文件后，单击【确定】按钮。

4) 加入主程序

现在只需要一个控制整个项目的主程序。在【项目管理器】窗口中单击【代码】选项卡，选中【程序】项，然后单击【新建】按钮，在弹出的【程序 1】窗口中输入下面的代码：

```

CLEAR SCREEN
oldpath = SET("path")
=SETPATH()
OPEN DATABASE  学生成绩
DO 主菜单.mpr
READ EVENTS
CLOSE DATABASE
SET SYSMENU TO DEFAULT
SET PATH TO &oldpath
RELEASE oldpath
FUNCTION SETPATH()
LOCAL lcSys16, lcProgram
lcSys16 = SYS(16)
lcProgram = SUBSTR(lcSys16, AT(":", lcSys16) - 1)
CD LEFT(lcProgram, RAT("\", lcProgram))
SET PATH TO PROGS, FORMS, MENUS, DATAS, BITMAPS, REPORTS, LIBS
ENDFUNC

```

关闭【程序 1】窗口，输入程序名“主程序”并选择“PROG”目录存放，扩展名“.prg”自动加入并返回到【项目管理器】窗口。使用鼠标右键单击，并在弹出的快捷菜单中选择【设置主文件】命令。设置完成后，“主程序”被加粗显示。此后 Visual FoxPro 便以“主程序.prg”来启动应用程序。现在的应用程序能完成下列任务：

- ① 保存 Visual FoxPro 原先的搜索路径，并设置应用程序的搜索路径。
- ② 打开数据库，使“教师表”和其他表的关系可用。
- ③ 使用创建的菜单取代 Visual FoxPro 的标准菜单。注意菜单是用它所生成的带有扩展名的代码如“主菜单.mpr”来表示的。从这时起，我们设计的菜单将是屏幕上的唯一菜单，直到使用 SET SYSMENU TO DEFAULT 命令。
- ④ 执行 READ EVENTS 命令。这样使表单和其他对象处于激活状态。命令在用户选择退出之前一直有效，EXIT 执行 CLEAR EVENTS 命令。此时，控制立即传递给 READ EVENTS 命令后的语句。

- ⑤ 从屏幕上移去所有遗留下来的表单。
- ⑥ 关闭所有用户文件。
- ⑦ 恢复原来的 Visual FoxPro 菜单和搜索路径。

## 5) 建立应用程序

在允许应用程序运行之前的最后一件事是生成它。单击位于【项目管理器】窗口右侧的【连编】按钮。这时将出现如图 9.16 所示的对话框。

【重新连编项目】操作读出应用程序的各种组成部分，建立项目文件，加入屏幕、程序和菜单中所引用的各种元素。可以在项目中只加入“主程序.prg”并让 Visual FoxPro 重建该项目，还可以发现其他的组成部分。

【连编应用程序】操作建立一个带有扩展名.app 的 Visual FoxPro 的输出文件。【连编可

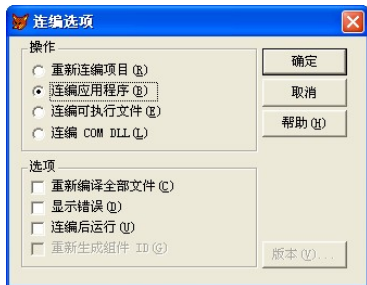


图 9.16 【连编选项】对话框

执行文件】操作将建立一个.exe 文件。它与.app 之间的区别在于：当运行一个.app 文件时，必须在 Visual FoxPro 的基础上才能运行它。但是，没有安装 Visual FoxPro 的计算机可以运行.exe 文件。如果生成了.exe 文件，那么运行这个程序只需点击程序名即可。【连编 COM DLL】操作使用项目文件中的类信息，创建一个带有.dll 文件扩展名的动态链接库。此处选择“连编应用程序”。

【重新编译所有文件】操作用于保证项目中所有的元素都被重新构造。当文件被编辑时，操作系统改变文件的日期/时间标记。项目管理器比较所有元素在项目文件中的日期和在目录中的日期。如果在目录中的日期更新，该文件将被重新编译。因此，如果另外一个开发人员修改了文件，但是他的系统的时钟不与原来人员的时钟同步，那么有可能他修改了文件，但是项目管理器认为不需要编译。

正常情况下，Visual FoxPro 把编译过程所遇到的错误放在一个与应用程序同名但扩展名为.err 的文件中。如果设置了“显示错误信息”，在构造应用程序的最后一步将打开一个编辑窗口显示错误信息。也可以不设置该选项，这样在错误出现时唯一能进行提示的，就是【项目】菜单上的【错误】项变得可用。选择【错误】项，将会打开错误窗口并显示编译错误。单击【确定】按钮，将项目文件命名为“学生成绩”，并选择“Visual FoxPro”目录存放。

## 6) 运行程序

要运行这个程序，可以从菜单中单击【程序】→【运行】命令并选择“学生成绩.app”，或者在【命令】窗口中输入“DO 学生成绩.app”。

可以看到，Visual FoxPro 的菜单变为一个新菜单，菜单中除了我们建立的两个菜单名之外，可能还有一个菜单名“项目”。关闭【项目管理器】窗口，此菜单名将消失。单击“教师表”，就可以查看到该表的内容。有兴趣的读者可以根据屏幕底部的图形按钮提示来操作一下。退出“教师表”屏幕后，单击“退出”菜单项，结束应用程序运行并恢复 Visual FoxPro 的系统菜单。

## 9.2.4 制作安装盘

Visual FoxPro 编译生成的.exe 文件是不能直接在另一台计算机上运行的，除非该计算机中已经装有 Visual FoxPro 系统，因为.exe 文件的运行要依赖于安装在 Windows 系统中的运行时刻库。为此需要为该软件制作一套安装盘，方法如下。

### 1. 建立子目录

在开发的软件的目录下建立一个子目录，比如命名为 VFP，当然也可以创建在其他位置或重新命名。

### 2. 添加项目内容

将该软件所要用到的数据库(.dbc)、数据库备注(.dct)、数据库索引(.dcx)、表(.dbf)、表索引(.cdx、.idx)、表备注(.fpt)、内存变量文件(.mem)等，再就是编译后的.exe 文件全部复制到上面所建的目录中。然后将复制的数据表中用来试运行的记录清除。但要注意有些数据可能是软件预先提供的，那么就不应该删除，比如在一个数据表中预先存入全国各省份名称与软件一起提供给用户，以免用户再去输入。



提示：源程序（.prg）文件、菜单文件、表单文件、报表文件、标签文件等不要复制，因为它们已经被编译在.exe文件中了；还有就是不属于软件运行的文件，如系统分析文件，也不要复制。

### 3. 启动Visual FoxPro系统并制作安装盘

如果 Visual FoxPro 系统已经启动，最好关闭所有打开的文件。具体制作步骤如下。

(1) 选择菜单上的【工具】→【向导】→【安装】命令，出现图 9.17 的定位文件对话框。单击【发布树目录】文本框的浏览按钮，找到在前面创建的目录，选定后单击【下一步】按钮，出现图 9.18 所示的指定组件对话框。

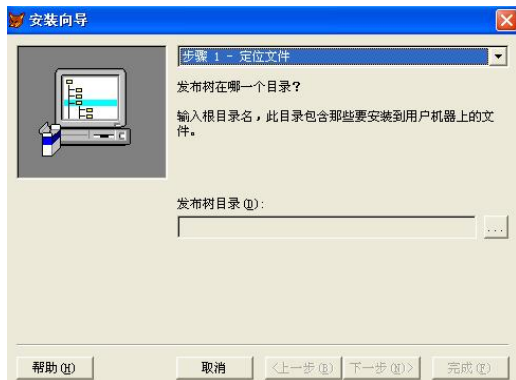


图 9.17 步骤 1-定位文件



图 9.18 步骤 2-指定组件

(2) 选择 Visual FoxPro 运行时刻组件，其他几个一般不选，单击【下一步】按钮，出现图 9.19 所示的磁盘映像对话框。

(3) 选择生成的安装文件存放的目录，一般放在软件目录中，即与.exe目录放在一起；还要选择安装方式，要么是 1.44 MB 软盘、要么是网络安装或两者都选。单击【下一步】按钮，出现图 9.20 所示的安装选项对话框。

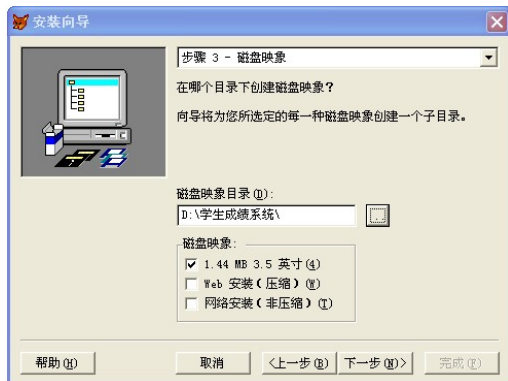


图 9.19 步骤 3-磁盘映像

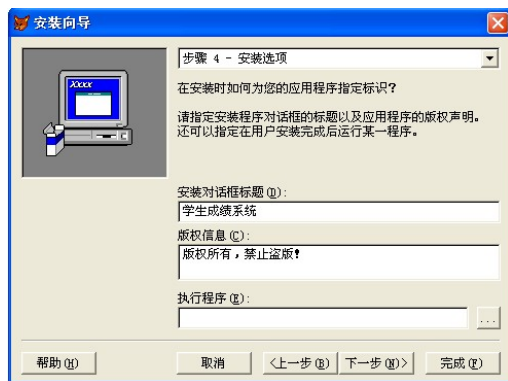


图 9.20 步骤 4-安装选项

(4) 在【安装对话框标题】和【版权信息】文本框中输入适当的内容。安装选项对话框主要提供的是安装软件时显示的信息，【版权信息】文本框中一定要输入内容。【执行程序】

文本框中不要输入内容，这里并不是指软件所要执行的程序。接着单击【下一步】按钮，出现图 9.21 所示的默认目标目录对话框。

(5) 输入安装的默认目录、开始菜单中的程序管理器组的名称；确定用户安装时是仅可以更改目录，还是目录与程序管理器组都可更改，一般就设为都可更改。单击【下一步】按钮，出现图 9.22 所示的改变文件设置对话框。

(6) 在文件列表中找到编译的.exe 或.app 文件，选中后面的【程序管理器】列中的小方框，出现【程序组菜单项】对话框，在【说明】文本框中输入在开始菜单中启动该软件的图标说明，在命令行中输入.exe文件名（或.app 文件名），记得在前面加上“%s\”，这是为了使安装在不同目录的软件也能正常运行。如果需要，还可为它选择一个图标（单击图标按钮选择，否则就显示为狐狸头），单击【确定】按钮，这样【程序管理器】列中的小方框就被选中。单击【下一步】按钮，出现图 9.23 所示的完成对话框。

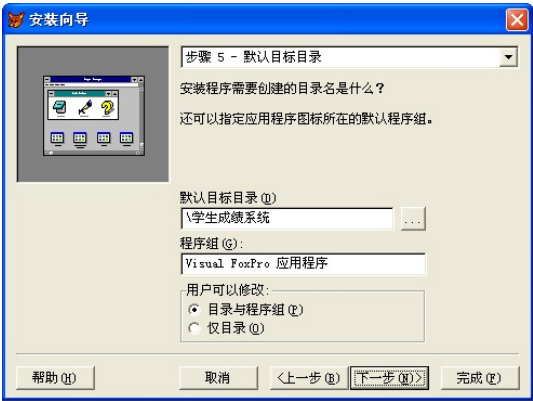


图 9.21 步骤 5-默认目标目录



图 9.22 步骤 6-改变文件设置

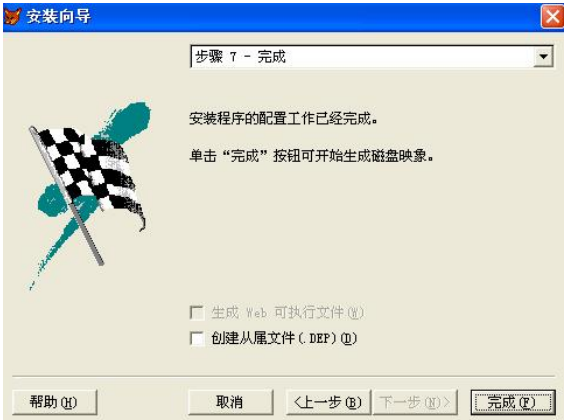


图 9.23 步骤 7-完成

如果没有问题就单击【完成】按钮开始制作安装盘。如果有问题，可以单击【上一步】按钮返回修改。单击【完成】按钮后，系统便开始按照设置制作安装盘，可能需要几分钟的时间，期间会有如图 9.24 的显示。制作完成后也会给出一个报告，如图 9.25 所示。



查看完报告后，单击【完成】按钮，随后可在磁盘上看到生成的安装文件目录，如果选择网络安装，目录是“netsetup”，其中包括安装软件所需的文件。如果选择软盘，目录则是“disk144”，其中还会有 disk1、disk2、disk3 等子目录，分别把每个目录的文件复制到一张软盘上，这样安装时从第一张盘开始，运行 setup 文件即可。

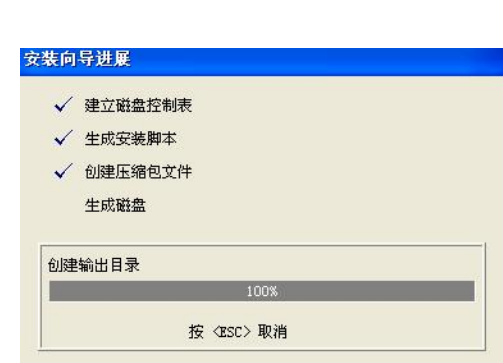


图 9.24 【安装向导进展】窗口



图 9.25 安装结果

练习与操作

练习 1 将数据库应用系统的文件制作成安装盘。参考步骤如下。

- (1) 在硬盘上创建一个发布树目录，并将要发布给用户的全部文件复制到该文件夹下。
- (2) 在硬盘上创建一个磁盘映象文件夹，用于保存制作的安装盘。
- (3) 执行菜单的【工具】→【向导】→【安装】命令，启动系统的【Visual FoxPro 安装向导】对话框。单击【创建目录】按钮，进入安装向导的“步骤 1”。
- (4) 在“步骤 1”的对话框中，单击【发布树目录】文本框后面的浏览按钮，并在弹出的【选择目录】对话框中选择在第(1)步已创建好的发布树文件夹。
- (5) 单击【选定】按钮返回【安装向导】对话框。
- (6) 单击【下一步】按钮，进入安装向导的“步骤 2”。选中【应用程序组件】中的【Visual FoxPro 运行时刻组件】复选框，然后单击【下一步】按钮进入安装向导的“步骤 3”。
- (7) 单击【磁盘映象目录】文本框后面的浏览按钮，并在弹出的【选择目录】对话框中选择在第(2)步已创建好的磁盘映象文件夹，然后单击【选定】按钮返回【安装向导】对话框。选中【磁盘映象】组中的【1.44 MB 3.5 英寸】复选框。
- (8) 单击【下一步】按钮，进入安装向导的“步骤 4”。
- (9) 分别在【安装对话框标题】和【版权信息】文本框中输入文字内容。单击【执行程序】文本框后面的浏览按钮，并在弹出的【打开】对话框中选择希望用户在安装应用程序之后立即执行的应用程序文件名。
- (10) 单击【下一步】按钮，进入安装向导的“步骤 5”。
- (11) 在【默认目标目录】文本框后面输入应用程序要安装的目标文件夹。
- (12) 单击【完成】按钮，在等待系统创建应用程序磁盘映象过程结束后，安装盘即制作完成。

## 思考与练习 9

### 一、思考题

1. 在 Visual FoxPro 中, 系统开发应用程序的基本步骤有哪些?
2. 连编应用程序的过程是什么?
3. 应用程序生成器的功能是什么?

### 二、选择题

1. 在 Visual FoxPro 中, 不可以通过\_\_\_\_\_方法来创建应用程序。  
A. 项目管理器                      B. 菜单                      C. 数据工作期                      D. “命令”窗口
2. 常用的程序文件扩展名有 .exe、.app 和 \_\_\_\_\_3 种。  
A. .prg                      B. .dll                      C. .com                      D. .pjt
3. 下面有关应用系统运行的描述正确的是\_\_\_\_\_。  
A. .app 文件可以在 Visual FoxPro 和 Windows 环境下运行  
B. .app 文件只能在 Windows 环境下运行  
C. .exe 文件可以在 Visual FoxPro 和 Windows 环境下运行  
D. .exe 文件只能在 Windows 环境下运行
4. 通过连编可以生成多种类型的文件, 但是却不能生成\_\_\_\_\_。  
A. .prg 文件                      B. .app 文件                      C. .dll 文件                      D. .exe
5. 如果将一个表单文件设置为“排除”状态, 那么它\_\_\_\_\_。  
A. 不参加连编                      B. 排除在应用程序外  
C. 本次不编译                      D. 不显示编译错误
6. 在应用程序生成器的【常规】选项卡中, 选择“正常”应用程序类型, 即生成将在 Visual FoxPro 主窗口中运行的\_\_\_\_\_。  
A. .app 应用程序                      B. .exe 应用程序                      C. COM DLL 文件                      D. .prg 文件
7. 在 Visual FoxPro 中, 为了对程序中的引用进行校验, 需要进行\_\_\_\_\_操作。  
A. 重新连编项目                      B. 连编应用程序                      C. 连编可执行文件                      D. 连编 COM DLL
8. 在主程序设计中, 主文件或主应用程序对象应该做的第一件事情是\_\_\_\_\_。  
A. 显示初始用户界面                      B. 对应用程序的环境进行初始化  
C. 制定控制事件循环的条件                      D. 组织主程序文件
9. 下列各项中不属于应用程序框架的功能的是\_\_\_\_\_。  
A. 显示菜单和工具栏  
B. 提供启动和清理程序  
C. 帮助开发者确定应用程序的功能、用户输入数据的方式、应用程序的外观等  
D. 设计表单和报表的外观样式
10. 利用应用程序生成器的【报表】选项卡不可以进行的操作是\_\_\_\_\_。  
A. 指定选中报表的名称                      B. 删除选中报表                      C. 修改选中报表                      D. 创建简单报表

### 三、填空题

1. 所谓程序, 简单地讲就是\_\_\_\_\_的集合。

2. 利用 Visual FoxPro 的“应用程序生成器”进行连编项目时，如果出现某个程序正在使用，不能编译的错误，可以先执行\_\_\_\_\_命令，然后重新连编。
3. 若要创建一个程序文件 PRL，应在【命令】窗口中输入\_\_\_\_\_命令。程序文件的扩展名为\_\_\_\_\_。
4. 在 Visual FoxPro 中利用应用程序生成器设计应用程序时，为了使所做的修改与当前活动项目保持一致，可以单击【高级】选项卡中的\_\_\_\_\_按钮。
5. 在 Visual FoxPro 中，可以利用\_\_\_\_\_指令来对应用系统的环境进行初始化。
6. 在 Visual FoxPro 中，如果一个文件是只读文件，那么此文件应标记为\_\_\_\_\_。
7. 在项目管理器的\_\_\_\_\_选项卡中可以查看和修改表单与报表。
8. 在开发应用程序时，可以利用\_\_\_\_\_将应用程序的各个部分组织起来。
9. 通过\_\_\_\_\_，Visual FoxPro 能够分析文件的引用，然后重新编译过期的文件。
10. 应用程序生成器的\_\_\_\_\_选项卡中指定了应用程序的产品信息。

## 第 10 章 小型系统项目开发

### 10.1 教职工信息管理系统

#### 10.1.1 案例设计的提出及要解决的问题

学校的教职工信息管理包括对教职工基本情况、教职工工资情况及部门工资情况等的管理。通过建立“教职工信息管理系统”这样一个基于 VFP 开发的小型数据库应用系统，实现对教职工基本信息和工资信息的输入、查询、维护及输出等功能。

#### 10.1.2 案例设计要实现的主要功能

以下建立的教职工信息管理系统，包含了三个有相互关系的表，并设置了各个表中某些字段的属性；通过建立不同类型的视图，实现了对数据的浏览、统计和添加等操作；通过建立一个主菜单将所要实现的各种功能，用创建的不同类型的表单实现与用户进行交互操作的友好界面；通过建立报表，实现了对选课成绩等数据的分析整理与输出；最后通过建立主程序和项目的连编，形成一个可执行的教职工信息管理系统，达到了快速完成教职工信息查询、工资管理、信息录入等操作的效果。

##### 1. 系统登录

系统登录成功后即可进入系统菜单，系统菜单包括：“数据建立”、“计算查询”、“工资报表”和“退出系统”。“数据建立”子菜单包括“基本情况”和“工资情况”菜单项。“计算查询”子菜单包括“基本情况查询”和“工资计算汇总”菜单项。

##### 2. 数据建立

输入基本情况、工资情况和部门工资表等信息。其中，基本情况信息输入表单和工资情况信息输入表单中采用用户命令按钮组来实现记录的移动。

##### 3. 计算查询

① 基本情况查询：根据姓名和部门名称查询相应的记录在表格控件中显示；用工具栏选择显示对应当前记录的基本情况、工资情况和部门情况的信息。

② 工资计算汇总：在表单用一个表格控件显示工资情况，用另一个表格控件显示部门工资汇总情况。

③ 工资报表：报表显示工资情况，列出每一个部门的工资明细信息之后将汇总部门的工资总和。

10.1.3 案例设计操作思路

“教职工信息管理系统”案例设计的基本思路如下所示。

- ① 编写一个程序文件 MAIN.prg 作为项目的主文件，由它调用用户登录表单“封面.scx”。
- ② 用户登录成功后，由用户登录表调用系统菜单“主菜单.mnx”。在系统菜单中操作数据库及其表的表单、查询和报表。
- ③ 在表单中分别处理教职工基本表、教职工工资表和部门工资表。
- ④ 通过视图将教职工基本表、教职工工资表关联起来。报表输出工资数据并进行计算汇总。

10.1.4 案例设计参考步骤

1. 程序的总体结构设计

在开始程序设计之前，应先将程序的总体结构以层次图的形式表示出来，便于对程序分层编程和实现，如图 10.1 所示。

图的第一层为系统层，通常对应主程序；第二层为子系统层，一般起划分系统功能的作用，通常对应主菜单；第三层为功能层，对应菜单项，一般通过表单的调用实现某些特定的功能。

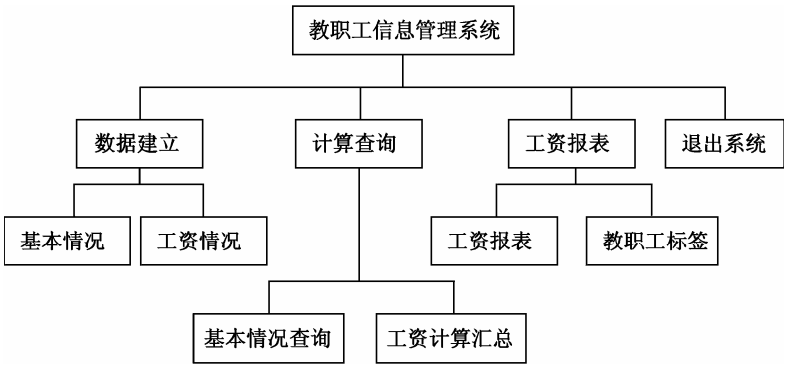


图 10.1 案例设计结构

2. 表

根据分析，确定系统要设置如下表。

- ① 教职工基本表，包括字段：编号、姓名、性别、职称、出生日期、文化程度、婚否、工资、照片、备注。编号为主索引。
  - ② 教职工工资表，包括字段：编号、姓名、部门名称、基本工资、津贴工资、职贴工资、其他工资、应发工资、扣款、实发工资。编号为主索引，部门名称为普通索引。
  - ③ 部门工资表，包括字段：部门编号、人数、部门名称、基本工资、津贴工资、职贴工资、其他工资、应发工资、扣款、实发工资。部门名称为主索引。
- 各表的表结构及记录如图 10.2~图 10.4 和表 10.1~表 10.3 所示。

表 10.1 教职工基本表

| 字段名  | 类型  | 宽度 | 小数位数 | 索引  |
|------|-----|----|------|-----|
| 编号   | 字符型 | 5  |      | 主索引 |
| 姓名   | 字符型 | 6  |      |     |
| 性别   | 字符型 | 2  |      |     |
| 职称   | 字符型 | 10 |      |     |
| 出生日期 | 日期型 | 8  |      |     |
| 文化程度 | 字符型 | 6  |      |     |
| 婚否   | 逻辑型 | 1  |      |     |
| 工资   | 数值型 | 6  | 1    |     |
| 照片   | 通用型 | 4  |      |     |
| 备注   | 备注型 | 4  |      |     |



| 编号   | 姓名  | 性别 | 职称  | 出生日期     | 文化程度 | 婚否 | 工资     | 照片  | 备注  |
|------|-----|----|-----|----------|------|----|--------|-----|-----|
| 5012 | 索武军 | 男  | 政工师 | 10/01/57 | 本科   | T  | 2587.7 | gen | gen |
| 1013 | 张德玉 | 女  | 教授  | 12/01/59 | 博士   | T  | 1234.5 | gen | gen |
| 2024 | 达林海 | 男  | 副教授 | 02/01/62 | 硕士   | T  | 1935.9 | gen | gen |
| 3035 | 杨涛寿 | 男  | 助经  | 09/01/77 | 大专   | F  | 1935.5 | gen | gen |
| 1046 | 王德兴 | 男  | 助教  | 03/01/69 | 本科   | F  | 1070.5 | gen | gen |
| 3047 | 王正直 | 女  | 讲师  | 09/01/69 | 本科   | T  | 1090.5 | gen | gen |
| 4046 | 单新荣 | 女  | 助教  | 09/01/70 | 硕士   | F  | 2920.6 | gen | gen |
| 5039 | 张光荣 | 男  | 工程师 | 12/01/67 | 本科   | T  | 2756.5 | gen | gen |
| 6041 | 杨小华 | 女  | 高工  | 12/01/62 | 本科   | F  | 3263.5 | gen | gen |
| 3012 | 朱平平 | 男  | 教授  | 10/01/63 | 硕士   | T  | 3934.5 | gen | gen |
| 2303 | 肖红  | 男  | 讲师  | 09/17/74 | 大专   | T  | 2733.5 | gen | gen |
| 6014 | 张月卿 | 男  | 助工  | 06/01/65 | 本科   | T  | 1263.5 | gen | gen |
| 1025 | 张宏艳 | 女  | 副教授 | 06/01/65 | 大专   | F  | 2721.5 | gen | gen |
| 4046 | 王琳秋 | 女  | 讲师  | 09/09/72 | 大专   | F  | 1919.6 | gen | gen |

图 10.2 “教职工基本表”的记录

表 10.2 教职工工资表

| 字段名  | 类型  | 宽度 | 小数位数 | 索引   |
|------|-----|----|------|------|
| 编号   | 字符型 | 5  |      | 主索引  |
| 姓名   | 字符型 | 6  |      |      |
| 部门名称 | 字符型 | 2  |      | 普通索引 |
| 基本工资 | 数值型 | 10 |      |      |
| 津贴工资 | 数值型 | 6  | 1    |      |
| 职贴工资 | 数值型 | 6  | 1    |      |
| 其他工资 | 数值型 | 6  | 1    |      |
| 应发工资 | 数值型 | 6  | 1    |      |
| 扣款   | 数值型 | 6  | 1    |      |
| 实发工资 | 数值型 | 6  | 1    |      |

| 教职工工资表 |     |      |        |        |        |       |        |      |        |  |
|--------|-----|------|--------|--------|--------|-------|--------|------|--------|--|
| 编号     | 姓名  | 部门名称 | 基本工资   | 津贴工资   | 职贴工资   | 其他工资  | 应发工资   | 扣款   | 实发工资   |  |
| 3012   | 紫武军 | 院办   | 2587.7 | 1250.0 | 1100.0 | 300.0 | 5217.7 | 67.5 | 5150.2 |  |
| 1013   | 张德玉 | 一系   | 1234.5 | 600.0  | 500.0  | 200.0 | 2534.5 | 45.6 | 2488.9 |  |
| 2024   | 达林海 | 二系   | 1935.9 | 800.0  | 700.0  | 300.0 | 3735.9 | 33.7 | 3702.2 |  |
| 3035   | 杨涛寿 | 院办   | 1935.5 | 900.0  | 800.0  | 200.0 | 3835.5 | 33.5 | 3802.0 |  |
| 1046   | 王德兴 | 一系   | 1070.5 | 400.0  | 300.0  | 200.0 | 1970.5 | 26.7 | 1943.8 |  |
| 3047   | 王正直 | 三系   | 1090.5 | 400.0  | 300.0  | 180.0 | 1970.5 | 27.3 | 1943.2 |  |
| 4048   | 单新荣 | 四系   | 2920.6 | 1850.0 | 1500.0 | 160.0 | 6430.6 | 68.9 | 6361.7 |  |
| 5039   | 张光荣 | 计算中心 | 2756.5 | 1750.0 | 1300.0 | 140.0 | 5946.5 | 69.3 | 5877.2 |  |
| 6041   | 杨小华 | 实验中心 | 3283.5 | 2200.0 | 2000.0 | 120.0 | 7603.5 | 73.5 | 7530.0 |  |
| 3012   | 朱平平 | 三系   | 3934.5 | 2900.0 | 2400.0 | 300.0 | 9534.5 | 78.5 | 9456.0 |  |
| 2303   | 肖 红 | 二系   | 2733.5 | 1700.0 | 1500.0 | 80.0  | 6013.5 | 70.4 | 5943.1 |  |
| 6014   | 张月卿 | 实验中心 | 1263.5 | 600.0  | 500.0  | 60.0  | 2423.5 | 26.5 | 2397.0 |  |
| 1025   | 张宏艳 | 一系   | 2721.5 | 1200.0 | 1000.0 | 40.0  | 4961.5 | 56.7 | 4904.8 |  |
| 4046   | 王琳秋 | 四系   | 1919.6 | 900.0  | 700.0  | 20.0  | 3539.6 | 45.6 | 3494.0 |  |

图 10.3 “教职工工资表”的记录

表 10.3 部门工资表

| 字段名  | 类型  | 宽度 | 小数位数 | 索引  |
|------|-----|----|------|-----|
| 部门编号 | 字符型 | 5  |      |     |
| 人数   | 数值型 | 3  |      |     |
| 部门名称 | 字符型 | 10 |      | 主索引 |
| 基本工资 | 数值型 | 8  | 1    |     |
| 津贴工资 | 数值型 | 8  | 1    |     |
| 职贴工资 | 数值型 | 8  | 1    |     |
| 其他工资 | 数值型 | 8  | 1    |     |
| 应发工资 | 数值型 | 8  | 1    |     |
| 扣款   | 数值型 | 8  | 1    |     |
| 实发工资 | 数值型 | 8  | 1    |     |

| 部门工资表 |    |      |          |         |         |        |          |         |          |
|-------|----|------|----------|---------|---------|--------|----------|---------|----------|
| 部门编号  | 人数 | 部门名称 | 基本工资     | 津贴工资    | 职贴工资    | 其他工资   | 应发工资     | 扣款      | 实发工资     |
| 01    | 10 | 院办   | 25877.0  | 22345.5 | 10234.5 | 7896.4 | 66153.4  | 12354.0 | 53799.4  |
| 02    | 45 | 一系   | 55552.5  | 48945.0 | 14987.0 | 6789.0 | 126273.5 | 13452.0 | 112821.5 |
| 03    | 38 | 二系   | 73564.2  | 59656.0 | 17897.0 | 4567.0 | 155884.2 | 18796.0 | 137088.2 |
| 04    | 59 | 三系   | 114194.5 | 89765.0 | 43587.0 | 5432.0 | 252958.5 | 48765.0 | 204193.5 |
| 05    | 18 | 四系   | 19269.0  | 6589.0  | 4587.0  | 1432.0 | 31857.0  | 10765.0 | 21092.0  |
| 06    | 28 | 计算中心 | 30534.0  | 13456.5 | 9876.0  | 5674.0 | 59540.5  | 12543.0 | 46997.5  |
| 07    | 32 | 实验中心 | 93459.2  | 59667.0 | 29876.0 | 6754.0 | 189956.2 | 34765.0 | 155191.2 |

图 10.4 “部门工资表”的记录

3. 创建“教职工信息管理”的项目管理器

下面创建“教职工信息管理”的项目管理器。首先创建系统的工作文件夹为“d:\JZG”，然后创建“教职工信息管理”的项目管理器，项目文件名为“教职工信息管理.pjx”。打开项目文件“教职工信息管理.pjx”的项目管理器，如图 10.5 所示。

## 4. 创建教职工信息数据库及表

下面创建教职工信息数据库和相关的表，具体步骤如下。

(1) 创建名为“教职工信息库”的数据库。打开“教职工信息管理”的项目管理器，在该项目管理器的【数据】选项卡中新建数据库“教职工信息库”，并打开该数据库文件。

(2) 在数据库“教职工信息库”中分别创建表：“教职工基本表”、“教职工工资表”和“部门工资表”。表结构和记录如前所述。

(3) 建立表的关联。在项目管理器中选择“教职工信息库”数据库，单击【修改】按钮，建立下列永久关系：

- “教职工基本表”的主索引“编号”和“教职工工资表”的主索引“编号”之间建立一对一的关联。
- “部门工资表”的主索引“部门名称”和“教职工工资表”的普通索引“部门名称”之间建立一对多的关联。建立永久关系后的数据库如图 10.6 所示。



图 10.5 “教职工信息管理”的项目管理器

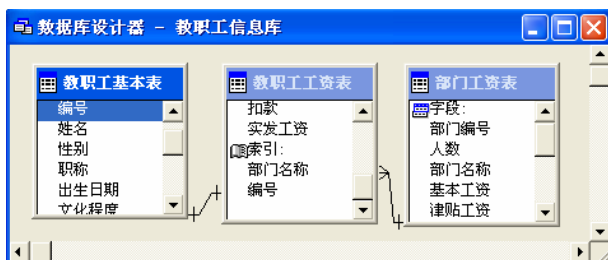


图 10.6 “教职工信息库”中表的关系

## 5. 创建视图

创建视图的具体步骤如下。

(1) 选择【视图】→【本地视图】命令，单击【新建】按钮，打开【视图设计器】窗口。

(2) 加入数据环境。在数据环境中加入“教职工工资表”和“部门工资表”。

(3) 设计输出字段，包括：

- “教职工工资表”的所有字段。
- “部门工资表”的“部门编号”字段。

(4) 选取连接条件：LEFT(教职工工资表.部门名称, 2)=部门工资表.部门名称。

(5) 指定排序依据：教职工工资表.编号。

(6) 指定更新条件：“教职工工资表”的所有数值字段（实发工资除外）。“教职工工资表.编号”为关键字段。选中【发送 SQL 更新】项，如图 10.7 所示。

(7) 生成的 SQL 语句：



```
SELECT  教职工工资表.编号, 教职工工资表.姓名, 部门工资表.部门编号, ;
教职工工资表.部门名称, 教职工工资表.基本工资, 教职工工资表.津贴工资, ;
教职工工资表.职贴工资, 教职工工资表.其他工资, 教职工工资表.应发工资, ;
教职工工资表.扣款, 教职工工资表.实发工资;
FROM  教职工信息库!部门工资表  INNER JOIN  教职工信息库!教职工工资表 ;
ON  部门工资表.部门名称 = 教职工工资表.部门名称;
ORDER BY  教职工工资表.编号
```

- (8) 预览结果视图，如图 10.8 所示。
- (9) 保存视图，命名为“GZST”。

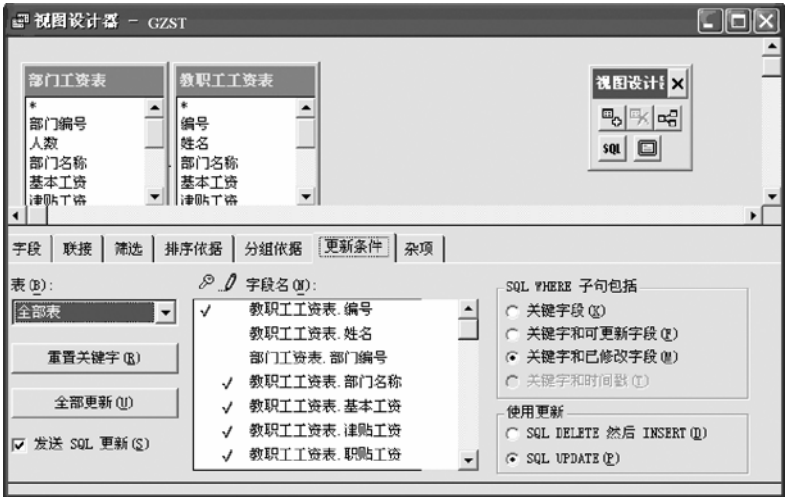


图 10.7 视图更新条件

| 编号   | 姓名  | 部门编号 | 部门名称 | 基本工资   | 津贴工资   | 职贴工资   | 其他工资  | 应发工资   | 扣款   | 实发工资   |
|------|-----|------|------|--------|--------|--------|-------|--------|------|--------|
| 1013 | 张德玉 | 02   | 一系   | 1234.5 | 600.0  | 500.0  | 200.0 | 2534.5 | 45.6 | 2488.9 |
| 1025 | 张宏艳 | 02   | 一系   | 2721.5 | 1200.0 | 1000.0 | 40.0  | 4961.5 | 56.7 | 4904.8 |
| 1046 | 王德兴 | 02   | 一系   | 1070.5 | 400.0  | 300.0  | 200.0 | 1970.5 | 26.7 | 1943.8 |
| 2024 | 达林海 | 03   | 二系   | 1935.9 | 800.0  | 700.0  | 300.0 | 3735.9 | 33.7 | 3702.2 |
| 2303 | 肖红  | 03   | 二系   | 2733.5 | 1700.0 | 1500.0 | 80.0  | 6013.5 | 70.4 | 5943.1 |
| 3012 | 索武军 | 01   | 院办   | 2567.7 | 1250.0 | 1100.0 | 300.0 | 5217.7 | 67.5 | 5150.2 |
| 3015 | 朱平平 | 04   | 三系   | 3934.5 | 2900.0 | 2400.0 | 300.0 | 9534.5 | 78.5 | 9456.0 |
| 3035 | 杨涛寿 | 01   | 院办   | 1935.5 | 900.0  | 800.0  | 200.0 | 3835.5 | 33.5 | 3802.0 |
| 3047 | 王正直 | 04   | 三系   | 1090.5 | 400.0  | 300.0  | 100.0 | 1970.5 | 27.3 | 1943.2 |
| 4046 | 王琳秋 | 05   | 四系   | 1919.6 | 900.0  | 700.0  | 20.0  | 3539.6 | 45.6 | 3494.0 |
| 4048 | 单新荣 | 05   | 四系   | 2920.6 | 1850.0 | 1500.0 | 160.0 | 6430.6 | 68.9 | 6361.7 |
| 5039 | 张光荣 | 06   | 计算中心 | 2758.5 | 1750.0 | 1300.0 | 140.0 | 5948.5 | 69.3 | 5879.2 |
| 6014 | 张月卿 | 07   | 实验中心 | 1263.5 | 600.0  | 500.0  | 60.0  | 2423.5 | 26.5 | 2397.0 |
| 6041 | 杨小华 | 07   | 实验中心 | 3283.5 | 2200.0 | 2000.0 | 120.0 | 7603.5 | 73.5 | 7530.0 |

图 10.8 视图预览结果

6. 创建表单

主菜单下的大部分功能都是通过表单来实现的。除了系统的登录界面，还包括“教职工基本情况.scx”、“教职工工资.scx”、“部门工资.scx”等表单。表单设计步骤如下。

- (1) 创建教职工基本情况信息录入表单“教职工基本情况.scx”，如图 10.9 所示。

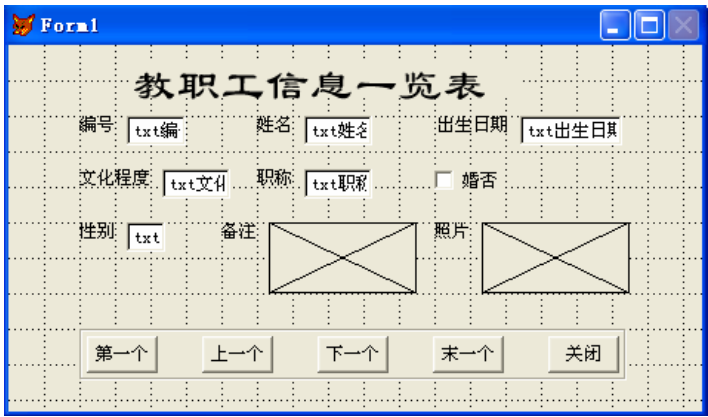


图 10.9 “教职工基本情况”表单

将“教职工基本表”加入表单数据环境，从数据环境中将“教职工基本表”中的各字段拖曳到表单上。在表单下部加入 1 个命令按钮组控件，在其中包含 5 个命令按钮，Caption 分别为“第一个”、“上一个”、“下一个”、“末一个”、“关闭”。编写相关的 Click 事件代码。

- 双击“第一个”命令按钮，输入其 Click 事件代码：

```
IF .NOT. EOF( )
    GO TOP
ENDIF
Thisform.Refresh
```

- 双击“上一个”命令按钮，输入其 Click 事件代码：

```
IF .NOT. BOF( )
    skip-1
ENDIF
Thisform.Refresh
```

- 双击“下一个”命令按钮，输入其 Click 事件代码：

```
IF .NOT. EOF( )
    SKIP
ENDIF
Thisform.Refresh
```

- 双击“末一个”命令按钮，输入其 Click 事件代码：

```
IF .NOT. EOF( )
    GO BOTTOM
ENDIF
Thisform.Refresh
```

- 双击“关闭”命令按钮，输入其 Click 事件代码：

```
RELEASE Thisform
```

(2) 创建教职工工资信息录入表单“教职工工资.scx”，如图 10.10 所示。

将“教职工工资表”加入表单数据环境，从数据环境中将教职工工资表中的各字段拖曳到表单上。在表单下部加入 1 个命令按钮组控件，在其中包含 5 个命令按钮，Caption 分别为“第一个”、“上一个”、“下一个”、“末一个”、“关闭”。编写 Click 事件代码，同上。

(3) 创建部门工资信息表单“部门工资.scx”，如图 10.11 所示。

(4) 创建教职工基本信息查找表单“查询表单.scx”，如图 10.12 所示。

图 10.10 “教职工工资”表单

图 10.11 “部门工资”表单

将“教职工基本表”、“教职工工资表”和“部门工资表”加入表单数据环境；在数据环境中建立下列关系：

- “教职工基本表. 编号”字段与“教职工工资表”的“编号”进行索引。
- “部门工资表. 部门编号”字段与“教职工基本表”的“部门编号”进行索引。

将“部门”组合框显示的行源 (RowSource) 设为“部门工资表”的“部门名称”字段。将表格的记录源 (RecordSource) 设为“教职工基本表”。

编写“确定”命令按钮的 Click 事件代码如下：

```
IF !EMPTY(This.Parent.Combo1.Value)
    SET FILTER TO 教职工基本表.姓名=TRIM(This.Parent.Combo1.Value)
    This.Parent.Refresh
    RETURN
ENDIF
IF !EMPTY(This.Parent.Value)
    bm=TRIM(This.Parent.Combo2.Value)
    SELECT 部门工资表
    LOCATE FOR 部门名称 = bm
    bmbh = 部门编号
    SELECT 教职工基本表
    SET FILTER TO LEFT(教职工基本表.编号,2) = bmbh
```

图 10.12 用于查询的表单

```
This.Parent.Refresh
RETURN
ENDIF
```

设置属性，编写代码，实现教职工基本信息的录入功能。

(5) 创建工资信息查询表单“工资信息.scx”，如图 10.13 所示。

将“教职工工资表”和“部门工资表”加入表单数据环境。表格 Grid1 的记录源(Record-Source)为前面创建的视图 GZBM。

- 编写“计算部门工资”命令按钮的 Click 事件代码：

```
DO FORM 部门工资表.scx
Thisform.Refresh
```

- 编写“关闭”命令按钮的 Click 事件代码：

```
Thisform.Release
```

(6) 创建系统登录表单“密码检验.scx”，如图 10.14 所示。

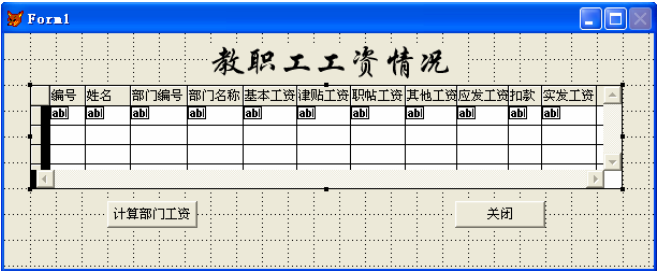


图 10.13 “工资信息”表单



图 10.14 “密码检验”表单

- 编写“确定”命令按钮的 Click 事件代码：

```
IF Thisform.Text1.Value="STU" AND Thisform.Text2.Value="abc123"
DO 主菜单.mpr
ELSE
MESSAGEBOX("密码或用户名错误",0,"提示")
ENDIF
```

- 编写“退出”命令按钮的 Click 事件代码：

```
Thisform.Release
CLEAR EVENTS
```

7. 创建菜单

创建菜单的步骤如下。

- (1) 在【项目管理器】窗口的【其他】选项卡中，选择【菜单】项，单击【新建】按钮。打开菜单设计器，设计应用系统菜单。应用系统菜单分成两级，生成的菜单文件为“主菜单.mnx”。
- (2) 设计应用系统主菜单。应用系统主菜单如图 10.15 所示。其中工资报表调用命令为 REPORT FORM E-GZBB PREVIEW。

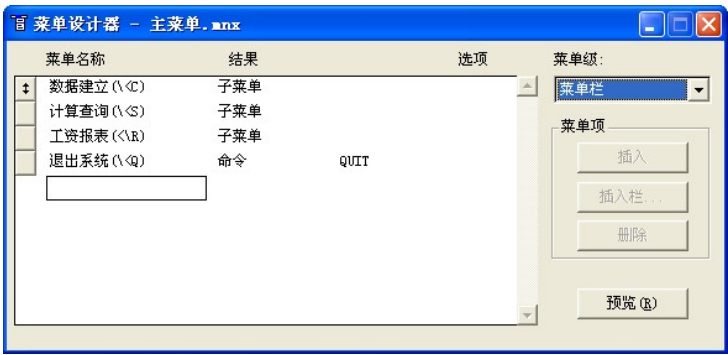


图 10.15 应用系统主菜单

(3) 设计“数据建立”子菜单，如图 10.16 所示。



图 10.16 “数据建立”子菜单

(4) 设计“计算查询”子菜单，如图 10.17 所示。



图 10.17 “计算查询”子菜单

### 8. 创建报表

一个应用系统一般都有报表作为系统的输出，例如创建一个工资报表，报表格式与实际输出格式类似。以“教职工工资表”为数据源，创建“工资报表.frx”，如图 10.18 所示。

在【项目管理器】窗口的【文档】选项卡中，选择【报表】项，单击【新建】按钮。打开报表设计器，设计工资报表。生成的报表文件为“工资报表.frx”。注意以下几点：

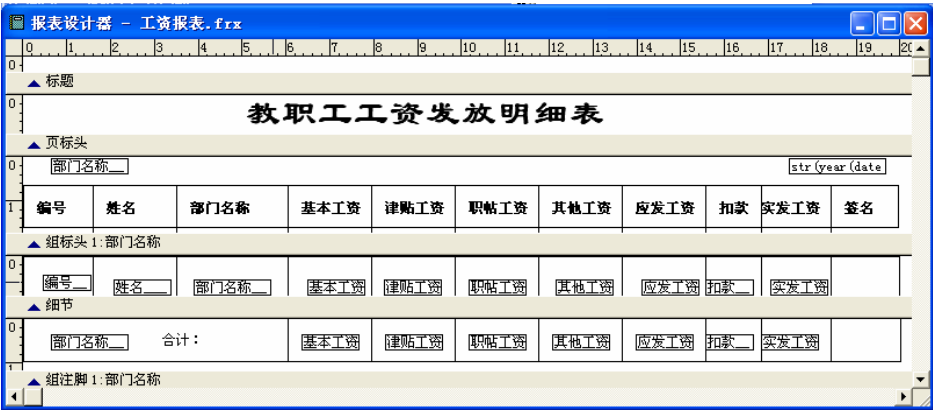


图 10.18 “工资报表”的报表设计器

- 在报表数据环境中加入“教职工工资表”。
- 分组表达式为 LEFT(教职工工资表.编号, 2)。
- 组标头：“编号”右边为文本框，表达式为“教职工工资表.部门名称”；右边报表日期为文本框，表达式为：STR(YEAR(DATE()), 4) + "年"+STR(MONTH(DATE()), 2)+ "月"。
- 对于细节和组注释中的文本框，直接从报表数据环境中将“教职工工资表”的字段拖曳过来即可。然后在组注释文本框上双击，选择【计算】按钮，在【计算字段】对话框中选中【总和】选项。
- 创建的“工资报表.frx”的运行结果如图 10.19 所示。

| 教职工工资发放明细表 |     |          |        |        |        |       |        |       |        |    |
|------------|-----|----------|--------|--------|--------|-------|--------|-------|--------|----|
| 二系         |     | 2009年 2月 |        |        |        |       |        |       |        |    |
| 编号         | 姓名  | 部门名称     | 基本工资   | 津贴工资   | 职帖工资   | 其他工资  | 应发工资   | 扣款    | 实发工资   | 签名 |
| 2024       | 达林海 | 二系       | 1935.9 | 800.0  | 700.0  | 300.0 | 3735.9 | 33.7  | 3702.2 |    |
| 2303       | 肖 红 | 二系       | 2733.5 | 1700.0 | 1500.0 | 80.0  | 6013.5 | 70.4  | 5943.1 |    |
| 二系 合计：     |     |          | 4669.4 | 2500.0 | 2200.0 | 380.0 | 9749.4 | 104.1 | 9645.3 |    |
| 计算中心       |     | 2009年 2月 |        |        |        |       |        |       |        |    |
| 编号         | 姓名  | 部门名称     | 基本工资   | 津贴工资   | 职帖工资   | 其他工资  | 应发工资   | 扣款    | 实发工资   | 签名 |
| 5039       | 张光荣 | 计算中心     | 2756.5 | 1750.0 | 1300.0 | 140.0 | 5946.5 | 69.3  | 5877.2 |    |
| 计算中心 合计：   |     |          | 2756.5 | 1750.0 | 1300.0 | 140.0 | 5946.5 | 69.3  | 5877.2 |    |

图 10.19 “工资报表”运行结果

接下来，以“教职工基本表”为数据源，创建“教职工标签.lbx”，如图 10.20 所示。

9. 创建程序

一般来说，应用系统需要很多程序文件，下面创建一个程序文件作为该项目的主文件。

在【项目管理器】窗口的【代码】选项卡中，选择【程序】项，单击【新建】按钮。打开程序设计器，设计系统程序。生成的程序文件为“主程序.prg”，其程序代码如下：

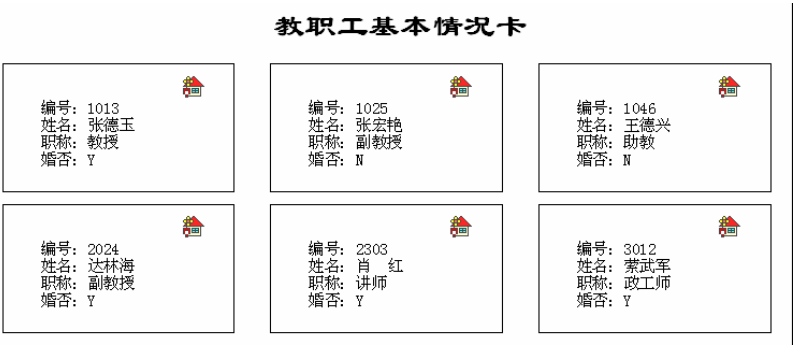


图 10.20 教职工标签

```
PUBLIC  uname
CLEAR
SET TALK OFF
SET DATE TO ymd
SET CENT ON
SET DEFA TO d:\jzg
DO FORM 封面.scx
READ EVENTS
CLEAR
CLOSE ALL
```

10. 编译应用程序

打开项目文件“教职工信息管理.pjx”，并在连编生成“教职工信息管理系统.exe”文件后运行。具体步骤如下。

- (1) 单击【项目管理器】窗口中的【连编】按钮，弹出【连编选项】对话框。
- (2) 在对话框中选择【连编可执行文件】单选项，然后单击【确定】按钮。
- (3) 在弹出的【另存为】对话框中，选择适当的存储路径，并在【文件名】文本框中输入“教职工信息管理系统”。
- (4) 然后单击【保存】按钮。

单击【项目管理器】窗口右上角的关闭按钮，关闭项目设计器。执行【程序】→【运行】命令，并在弹出的【运行】对话框中双击“教职工信息管理系统.exe”，即可进入该数据库应用系统主界面。

11. 运行应用程序

- (1) 双击应用程序文件“主程序.prg”，系统显示登录表单，如图 10.21 所示。
- (2) 5 秒钟后系统进入“密码检验”界面，输入登录名和口令，如果回答正确，单击【确定】按钮，进入主界面并显示应用系统菜单，如图 10.22 所示。
- (3) 选择“数据建立”→“基本情况”菜单项，系统运行“教职工基本情况.scx”表单，如图 10.23 所示。



图 10.21 登录表单运行结果



图 10.22 主界面中的应用系统菜单

- (4) 选择“工资情况”菜单项，系统运行“教职工工资.scx”表单，如图 10.24 所示。
- (5) 其他界面可一一显示。

|      |     |    |     |      |                                     |
|------|-----|----|-----|------|-------------------------------------|
| 编号   | 012 | 姓名 | 索武军 | 出生日期 | 10/01/1957                          |
| 文化程度 | 本科  | 职称 | 政工师 | 婚姻   | <input checked="" type="checkbox"/> |
| 性别   | 男   | 备注 | 照片  |      |                                     |

图 10.23 运行“教职工基本情况.scx”

|      |        |      |        |
|------|--------|------|--------|
| 编号   | 3012   | 姓名   | 索武军    |
| 基本工资 | 2567.7 | 津贴工资 | 1250.0 |
| 职贴工资 | 1100.0 | 其他工资 | 300.0  |
| 扣款   | 67.5   | 实发工资 | 5150.2 |

图 10.24 运行“教职工工资.scx”



## 10.2 图书信息管理系统

### 10.2.1 案例设计的提出及要解决的问题

图书信息管理包括对图书基本情况、图书借阅情况及图书库存情况等的管理。通过建立“图书信息管理系统”这样一个基于 VFP 开发的小型的数据库应用系统,实现对图书的基本信息输入、输出,查询图书的库存数量,以及维护图书的相关信息等功能。

### 10.2.2 案例设计要实现的主要功能

以下建立的图书信息管理系统,包含了三个有相互关系的表,并设置了各个表中某些字段的属性;通过建立不同类型的视图,实现了对数据的浏览、统计和添加等操作;通过建立一个主菜单将所要实现的各种功能,用创建的不同类型的表单实现与用户进行交互操作的友好界面;通过建立报表,实现了对图书资料等数据的分析整理与输出;最后通过建立主程序和项目的连编,形成一个可执行的图书信息管理系统,达到了快速完成图书信息查询、图书管理、新书录入等操作的效果。

#### 1. 系统登录

登录成功后即可进入系统菜单,其中:

- 主菜单包括“系统登录”、“图书查询”、“新书入库”、“新读者”、“借还书”和“退出系统”。
- “图书查询”子菜单包括“书名查询”、“书号查询”、“姓名查询读者”和“编号查询读者”菜单项。
- “借还书”子菜单包括“借书”和“还书”菜单项。

#### 2. 数据建立

设计输入图书信息表、读书表和借还书表信息。

### 10.2.3 案例设计操作思路

“图书信息管理系统”案例设计的思路有以下几点。

- ① 编写一个程序文件“主程序.prg”作为项目的主文件,由它调用用户登录表单“主界面.scx”。
- ② 用户登录成功后,由登录表单调用系统菜单“主菜单.mnx”。在系统菜单中调用数据库及其表单和报表等。
- ③ 表单中分别操作“图书信息表”、“读书表”和“借还书表”。

### 10.2.4 案例设计参考步骤

#### 1. 程序总体结构设计

在开始程序设计前,应先将程序的总体结构以层次图的形式表示出来,便于对程序进行分层编程和实现,如图10.25所示。图的第一层为系统层,通常对应主程序;第二层为子系统

层，一般起划分系统功能的作用，通常对应主菜单；第三层为功能层，对应菜单项，一般通过表单的调用实现某些特定的功能。

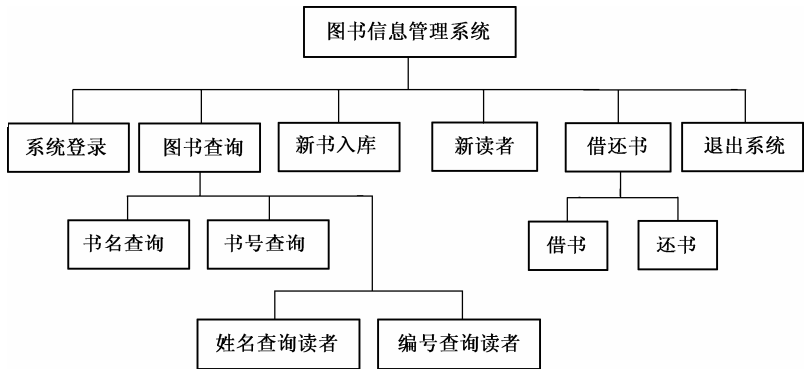


图 10.25 案例设计结构

2. 表

本案例根据分析确定系统要设置如下表。

- ① 图书信息表，包括字段：ISBN 号、书号、书名、作者、出版社、价格、简介。书号为主索引。
- ② 读书表，包括字段：读者编号、姓名、单位、身份证号、电话。读者编号为主索引。
- ③ 借还书表，包括字段：读者编号、书号、借书日期、还书日期、还书否。读者编号为普通索引，书号为候选索引。

各表表结构及记录如图 10.26~图 10.28 和表 10.4~表 10.6 所示。

表 10.4 图书信息表

| 字段名    | 类型  | 宽度 | 小数位数 | 索引  |
|--------|-----|----|------|-----|
| ISBN 号 | 字符型 | 10 |      |     |
| 书号     | 字符型 | 5  |      | 主索引 |
| 书名     | 字符型 | 24 |      |     |
| 作者     | 字符型 | 10 |      |     |
| 出版社    | 字符型 | 14 |      |     |
| 价格     | 数值型 | 4  | 2    |     |
| 简介     | 备注型 | 4  |      |     |

图 10.26 “图书信息表”的记录

表 10.5 读书表

| 字段名  | 类型  | 宽度 | 索引  |
|------|-----|----|-----|
| 读者编号 | 字符型 | 10 | 主索引 |
| 姓名   | 字符型 | 10 |     |
| 单位   | 字符型 | 26 |     |
| 身份证号 | 字符型 | 18 |     |
| 电话   | 字符型 | 12 |     |



| 读者编号  | 姓名  | 单位          | 身份证号               | 电话           |
|-------|-----|-------------|--------------------|--------------|
| 12011 | 张大民 | 大河电子技术有限公司  | 324110113204232655 | 86868686     |
| 12012 | 陈新华 | 百太原有限公司     | 640123453204357956 | 86868688     |
| 12013 | 李明明 | 无另计算机有限责任公司 | 210223333204308597 | 683818103    |
| 12014 | 无好远 | 巫山中学数学部     | 110111230958676347 | 123476895423 |

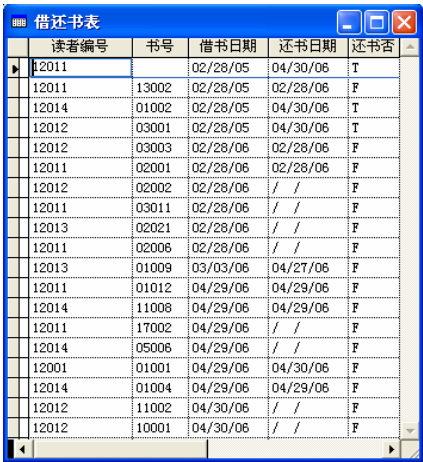
图 10.27 “读书表”的记录

表 10.6 借还书表

| 字段名  | 类型  | 宽度 | 索引   |
|------|-----|----|------|
| 读者编号 | 字符型 | 10 | 普通索引 |
| 书号   | 字符型 | 5  | 候选索引 |
| 借书日期 | 日期型 | 8  |      |
| 还书日期 | 日期型 | 8  |      |
| 还书否  | 逻辑型 | 1  |      |

3. 创建“图书管理”的项目管理器

首先创建系统的工作文件夹为“d:\图书管理”，然后创建“图书管理”的项目管理器，项目文件名为“图书管理.pjx”。打开项目文件“图书管理.pjx”的项目管理器，如图 10.29 所示。



| 读者编号  | 书号    | 借书日期     | 还书日期     | 还书否 |
|-------|-------|----------|----------|-----|
| 12011 |       | 02/28/05 | 04/30/06 | T   |
| 12011 | 13002 | 02/28/05 | 02/28/06 | F   |
| 12014 | 01002 | 02/28/05 | 04/30/06 | T   |
| 12012 | 03001 | 02/28/05 | 04/30/06 | T   |
| 12012 | 03003 | 02/28/06 | 02/28/06 | F   |
| 12011 | 02001 | 02/28/06 | 02/28/06 | F   |
| 12012 | 02002 | 02/28/06 | / /      | F   |
| 12011 | 03011 | 02/28/06 | / /      | F   |
| 12013 | 02021 | 02/28/06 | / /      | F   |
| 12011 | 02006 | 02/28/06 | / /      | F   |
| 12013 | 01009 | 03/03/06 | 04/27/06 | F   |
| 12011 | 01012 | 04/29/06 | 04/29/06 | F   |
| 12014 | 11008 | 04/29/06 | 04/29/06 | F   |
| 12011 | 17002 | 04/29/06 | / /      | F   |
| 12014 | 05006 | 04/29/06 | / /      | F   |
| 12001 | 01001 | 04/29/06 | 04/30/06 | F   |
| 12014 | 01004 | 04/29/06 | 04/29/06 | F   |
| 12012 | 11002 | 04/30/06 | / /      | F   |
| 12012 | 10001 | 04/30/06 | / /      | F   |

图 10.28 “借还书表”的记录

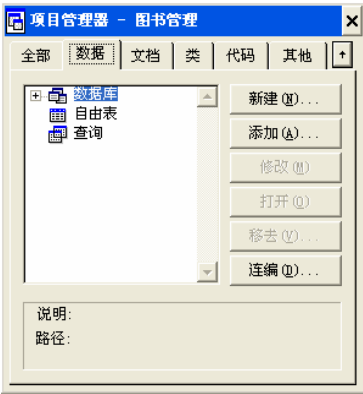


图 10.29 “图书管理”的项目管理器

4. 创建图书信息数据库及表

创建名为“图书管理库”的数据库。打开“图书管理”的项目管理器，在【数据】选项卡中新建数据库“图书管理库.dbc”。

打开数据库文件“图书管理库.dbc”，在数据库中分别创建表：图书信息表，读书表，借还书表。表结构和记录如前所述。

在项目管理器中选择“图书管理库”，单击【修改】按钮，建立下列永久关系：

- “图书信息表”的主索引“书号”和“借还书表”的主索引“书号”之间建立一对一的关联。
- “读书表”的主索引“读者编号”和“借还书表”的普通索引“读者编号”之间建立一对多的关联。

建立永久关系后的数据库如图 10.30 所示。

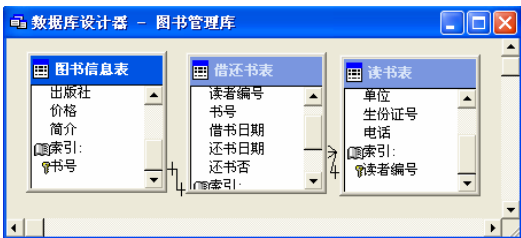


图 10.30 “图书管理库”中表的关系

5. 表单设计

主菜单下的大部分功能都是通过表单来实现的。除了系统的登录界面，所需表单还包括查询图书信息的表单、新书表表单、读书表表单和借还书情况表单等。

1) 创建查询图书信息的表单

查询图书信息所需的表单包括：“书名查询.scx”、“书号查询.scx”、“读者姓名查询.scx”、“读者编号查询.scx”。

① 书名查询.scx

(1) 将“图书信息表”加入表单数据环境，从数据环境中将“图书信息表”的各字段拖曳到表单上。

(2) 在表单上部加入 1 个标签和 1 个文本框，标签的 Caption 为“请输入查询书名”，编写文本框的 Keypress 事件代码：

```
LPARAMETERS nKeyCode, nShiftAltCtrl
IF nkeycode=13
    SELE 图书信息表
    LOCA FOR ALLT(This.Value)=ALLT(书名)
    IF FOUND()
        Thisform.Text2.Value=ALLT(isbn 号)
        Thisform.Text3.Value=ALLT(书名)
```

```

Thisform.Text4.Value=ALLT(书号)
Thisform.Text5.Value=ALLT(作者)
Thisform.Text6.Value=ALLT(str(价格))
Thisform.Text7.Value=ALLT(出版社)
Thisform.Edit1.Value=简介
ELSE
    MESSAGEBOX("对不起, 没有您要查询的书!")
ENDIF
ENDIF
Thisform.Refresh

```

- (3) 在表单下部加入 1 个命令按钮, Caption 为“退出”, 编写 Click 事件过程的代码:

```
Thisform.Release
```

“书名查询”表单如图 10.31 所示。



图 10.31 “书名查询”表单

## ② 书号查询.scx

(1) 将“图书信息表”加入表单数据环境, 从数据环境中将“图书信息表”的各字段拖曳到表单上。

(2) 在表单上部加入 1 个标签和 1 个文本框, 标签的 Caption 为“请输入查询书号”, 编写文本框的 Keypress 事件代码:

```

LPARAMETERS nKeyCode, nShiftAltCtrl
IF nkeycode=13
    SELE 图书信息表
    LOCA FOR ALLT(This.Value)=ALLT(书号)
    IF FOUND( )
        Thisform.Text2.Value=ALLT(isbn 号)
        Thisform.Text3.Value=ALLT(书名)
        Thisform.Text4.Value=ALLT(书号)
        Thisform.Text5.Value=ALLT(作者)
        Thisform.Text6.Value=ALLT(str(价格))
        Thisform.Text7.Value=ALLT(出版社)
        Thisform.Edit1.Value=简介
    ELSE

```

```

MESSAGEBOX("对不起, 没有您要查询的书!")
ENDIF
ENDIF
Thisform.Refresh

```

(3) 在表单下部加入 1 个命令按钮, Caption 为“退出”, 编写 Click 事件代码:

```
Thisform.Release
```

“书号查询”表单如图 10.32 所示。

图 10.32 “书号查询”表单

### ③ 读者姓名查询.scx

(1) 将“读书表”加入表单数据环境, 从数据环境中将“读书表”的各字段拖曳到表单上。

(2) 在表单上部加入 1 个标签和 1 个文本框, 标签的 Caption 为“请输入读者姓名”, 编写文本框的 Keypress 事件代码:

```

LPARAMETERS nKeyCode, nShiftAltCtrl
IF nkeycode=13
    SELE 读书表
    LOCA FOR ALLT(姓名)=ALLT(This.Value)
    IF FOUND( )
        Thisform.Text2.Value=ALLT(读者编号)
        Thisform.Text3.Value=ALLT(姓名)
        Thisform.Text4.Value=ALLT(单位)
        Thisform.Text5.Value=ALLT(身份证号)
        Thisform.Text6.Value=ALLT(电话)
    ELSE
        MESSAGEBOX("对不起, 没有您要查询的读者!")
    ENDIF
ENDIF
ENDIF
Thisform.Refresh

```

(3) 在表单下部加入 1 个命令按钮, Caption 为“退出”, 编写 Click 事件代码:

```
Thisform.Release
```

“读者姓名查询”表单如图 10.33 所示。

#### ④ 读者编号查询.scx

(1) 将“读书表”加入表单数据环境，从数据环境中将“读书表”的各字段拖曳到表单上。

(2) 在表单上部加入 1 个标签和 1 个文本框，标签的 Caption 为“请输入读者编号”，编写文本框的 Keypress 事件代码：

```
lparameters nKeyCode, nShiftAltCtrl
IF nkeycode=13
    SELE 读书表
    LOCA FOR ALLT(读者编号)=ALLT(This.Value)
    IF FOUND( )
        Thisform.Text2.Value=ALLT(读者编号)
        Thisform.Text3.Value=ALLT(姓名)
        Thisform.Text4.Value=ALLT(单位)
        Thisform.Text5.Value=ALLT(身份证号)
        Thisform.Text6.Value=ALLT(电话)
    ELSE
        MESSAGEBOX("对不起，没有您要查询的读者!")
    ENDIF
ENDIF
Thisform.Refresh
```

(3) 在表单下部加入 1 个命令按钮，Caption 为“退出”，编写 Click 事件代码：

```
Thisform.Release
```

“读者编号查询”表单如图 10.34 所示。

图 10.33 “读者姓名查询”表单

图 10.34 “读者编号查询”表单

## 2) 创建“新书表.scx”

“新书表”表单如图 10.35 所示，创建步骤如下。

(1) 利用表单向导创建“新书表.scx”。单击菜单栏上的【文件】→【新建】命令，在【新建】对话框中选择【表单】项，单击【向导】按钮。在【向导选取】对话框中选择【表单向导】，单击【确定】按钮。

(2) 在【表单向导】对话框的【数据库和表】列表框中选择想要的“图书信息表”，然后从【可用字段】列表框中将需要的全部字段移到【选定字段】列表框中，单击【下一步】按钮。

(3) 选定一种样式——标准式，单击【下一步】按钮。

(4) 从【可用的字段或索引标识】列表框中把“书号”字段添加到【选定字段】列表框中，选择“升序”。

(5) 选择【保存并运行表单】项，单击【完成】按钮。

图 10.35 “新书表”表单

### 3) 创建“读书表.scx”

“读书表”表单如图 10.36 所示，创建步骤如下。

(1) 利用表单向导创建“读书表.scx”。单击菜单栏上的【文件】→【新建】命令，在【新建】对话框中选择【表单】项，单击【向导】按钮。在【向导选取】对话框中选择【表单向导】，单击【确定】按钮。

(2) 在【表单向导】对话框的【数据库和表】列表框中选择想要的“读书表”，然后从【可用字段】列表框中将需要的全部字段移到【选定字段】列表框中，单击【下一步】按钮。

(3) 选定一种样式——标准式，单击【下一步】按钮。

(4) 从【可用的字段或索引标识】列表框中把“读者编号”字段添加到【选定字段】列表框中，选择“升序”。

(5) 选择【保存并运行表单】，单击【完成】按钮。

图 10.36 “读书表”表单



#### 4) 创建借还书情况的表单

与借还书情况相关的表单包括：“借书工作.scx”、“还书工作.scx”。各表单的创建如下所示。

##### ① 借书工作.scx

(1) 将“读书表”加入表单数据环境中。

(2) 在表单上部加入 7 个标签、2 个文本框、2 个命令按钮和 1 个表格控件, 标签的 Caption 及各控件布局如图 10.37 所示。

图 10.37 “借书工作”表单

(3) 编写文本框 Text1 的 Keypress 事件代码:

```
lparameters nKeyCode, nShiftAltCtrl
IF nkeycode=13
    SELE 读书表
    LOCAL FOR 读者编号=ALLT(this.value)
    IF !FOUN( )
        MESSAGEBOX("没有此人! 借书证号不正确!")
    ELSE
        Thisform.Label2.Caption="姓名: "+ALLT(姓名)
        a1=ALLT(读者编号)
        Thisform.Grid1.Recordsource="SELECT 图书信息表.书号, ;
            图书信息表.isbn 号,图书信息表.书名 FROM 借还书表,图书信息表;
            WHERE 借还书表.书号=图书信息表.书号 AND 借还书表.读者编号=a1;
            AND !借还书表.还书否 INTO curs zzk"
        Thisform.Refresh
    ENDIF
ENDIF
```

(4) 在表单下部加入 1 个命令按钮, Caption 为“退出”, 编写 Click 事件代码:

```
Thisform.Release
```

(5) 编写文本框 Text2 的 Keypress 事件代码:

```
lparameters nKeyCode, nShiftAltCtrl
```

```

IF nkeycode=13
    SELE 图书信息表
    LOCA FOR 书号=ALLT(this.value)
    IF !FOUN( )
        MESSAGEBOX("没有这本书!")
    ELSE
        Thisform.Label3.Caption="书号: "+ALLT(isbn 号)
        Thisform.Label4.Caption="书名: "+ALLT(书名)
        Thisform.Label5.Caption="作者: "+ALLT(作者)
        Thisform.Refresh
    ENDIF
ENDIF
ENDIF

```

(6) 编写命令按钮“确定”的 Click 事件代码。

```

INSERT INTO 借还书表(读者编号,书号,借书日期,还书否);
Values(Thisform.Text1.Value,Thisform.Text2.Value,DATE( ),.F.)
UPDATE sb SET zt=.F. WHERE sh=ALLT(Thisform.Text2.Value)
Thisform.Text1.Value=""
Thisform.Text2.Value=""
*Thisform.Label2.Caption="姓名"
Thisform.Label3.Caption="书号"
Thisform.Label4.Caption="书名"
Thisform.Label5.Caption="作者"
a1=ALLT(Thisform.Text1.Value)
Thisform.Grid1.Recordsource="SELECT 图书信息表.书号,图书信息表.isbn 号,;
    图书信息表.书名 FROM 借还书表,图书信息表;
WHERE 借还书表.书号=图书信息表.书号 AND 借还书表.读者编号=a1 AND;
!借还书表.还书否 INTO curs zzk"
Thisform.Refresh

```

(7) 编写命令按钮“退出”的 Click 事件代码:

```
Thisform.Release
```

② 还书工作.scx

(1) 将“读书表”加入表单数据环境中。

(2) 在表单上部加入 6 个标签、1 个文本框、2 个命令按钮, 标签的 Caption 及各控件布局如图 10.38 所示。

(3) 编写文本框 Text1 的 Keypress 事件代码:

```

LPARAMETERS nKeyCode, nShiftAltCtrl
IF nkeycode=13
    IF LEN(ALLT(Thisform.Text1.Value))=0
        MESSAGEBOX("请输入书号")
    ELSE
        SELECT 借还书表
        LOCAL FOR 书号=ALLT(Thisform.Text1.Value)AND !还书否
        IF FOUN( )

```



图 10.38 “还书工作”表单

```

a1=ALLT(读者编号)
a2=ALLT(书号)
SELECT 读书表
LOCAL FOR 读者编号=a1
Thisform.Label6.Caption="借书者: "+ALLT(姓名)
SELECT 图书信息表
LOCAL FOR 书号=a2
Thisform.Label3.Caption="书号: "+ALLT(isbn 号)
Thisform.Label4.Caption="书名: "+ALLT(书名)
Thisform.Label5.Caption="作者: "+ALLT(作者)
Thisform.Refresh
ELSE
    MESSAGEBOX("书号输入不对! 请重新输入")
ENDIF
ENDIF
ENDIF
ENDIF

```

#### (4) 编写命令按钮“确定”的 Click 事件代码:

```

SELE 借还书表
REPL 还书否 WITH .T.,还书日期 WITH DATE( )
Thisform.Text1.Value=" "
Thisform.Label6.Caption="借者: "
Thisform.Label3.Caption="书号: "
Thisform.Label4.Caption="书名: "
Thisform.Label5.Caption="作者: "
Thisform.Text1.SETFOCUS
Thisform.Refresh

```

#### (5) 编写命令按钮“退出”的 Click 事件代码。

```
Thisform.Release
```

### 5) 创建“系统登录.scx”

“系统登录”表单布局如图10.39所示，具体实现步骤如下。

#### (1) 编写命令按钮“确定”的 Click 事件代码:

```

IF a>=3
    QUIT
ENDIF
IF ALLT(Thisform.Text1.Value)=ALLT("wfy") AND;
ALLT(Thisform.Text2.Value)=ALLT("12345")
    MESSAGEBOX("欢迎您进入本系统!!!")
    SET SKIP OF pad a2 OF wfy .F.
    SET SKIP OF pad a1 OF wfy .F.
    SET SKIP OF pad a3 OF wfy .F.
    SET SKIP OF pad a4 OF wfy .F.
    SET SKIP OF pad a5 OF wfy .F.
    SET SKIP OF pad a6 OF wfy .F.
    SET SKIP OF pad a7 OF wfy .F.

```



图 10.39 “系统登录”表单

```

SET SKIP OF pad a8 OF wfy .F.
RELEASE Thisform
ELSE
SELE 密码表
LOCA FOR ALLT(Thisform.Text1.Value)=ALLT(xm)
IF FOUND( )
    IF ALLT(Thisform.Text2.Value)=ALLT(mm)
        SET SKIP OF pad a1 OF wfy .F.
        SET SKIP OF pad a2 OF wfy .F.
        SET SKIP OF pad a7 OF wfy .F.
        SET SKIP OF pad a8 OF wfy .F.
        IF qx="1"
            SET SKIP OF pad a2 OF wfy .F.
            SET SKIP OF pad a3 OF wfy .F.
            SET SKIP OF pad a4 OF wfy .T.
            SET SKIP OF pad a5 OF wfy .T.
            SET SKIP OF pad a6 OF wfy .F.
        ELSE
            IF qx="2"
                SET SKIP OF pad a3 OF wfy .F.
                SET SKIP OF pad a4 OF wfy .F.
                SET SKIP OF pad a4 OF wfy .T.
                SET SKIP OF pad a6 OF wfy .T.
            ELSE
                IF qx="3"
                    SET SKIP OF pad a3 OF wfy .T.
                    SET SKIP OF pad a4 OF wfy .T.
                    SET SKIP OF pad a5 OF wfy .F.
                    SET SKIP OF pad a6 OF wfy .T.
                ENDIF
            ENDIF
        ENDIF
    ENDIF
    RELEASE Thisform
ELSE
    a=a+1
    MESSAGEBOX("对不起, 您输入的用户名或密码错!" +STR(a,1)+"次")
ENDIF
ELSE
    a=a+1
    MESSAGEBOX("对不起, 您是非法用户, 请您马上离开, 否则····"+STR(a,1)+":次错误")
    Thisform.Refresh
    Thisform.Text1.Value=""
    Thisform.Text2.Value=""
ENDIF
ENDIF
ENDIF

```

(2) 编写命令按钮“退出”的 Click 事件代码：

```
Thisform.Release
CLEAR EVENTS
```

6. 创建菜单

在【项目管理器】窗口的【其他】选项卡中，选择【菜单】项，单击【新建】按钮。打开菜单设计器，设计应用系统菜单。应用系统菜单分成两级，生成的菜单文件为“主菜单.mnx”。

- 设计应用系统的主菜单，如图 10.40 所示。

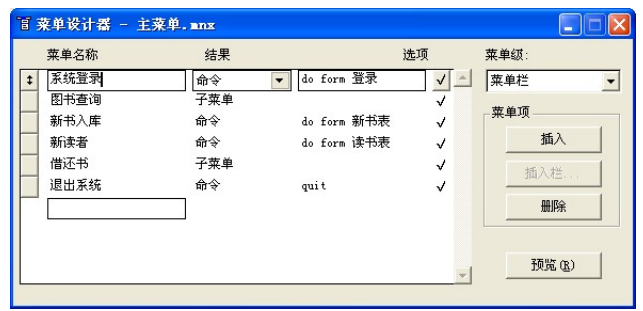


图 10.40 应用系统的主菜单

- 设计“图书查询”子菜单，如图 10.41 所示。

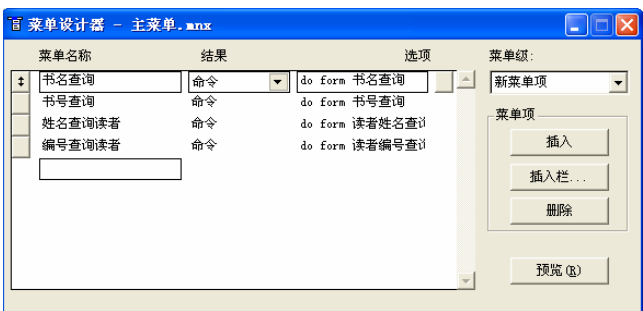


图 10.41 “图书查询”子菜单

- 设计“借还书”子菜单，如图 10.42 所示。



图 10.42 “借还书”子菜单

## 7. 创建程序

一般来说，一个应用系统需要很多程序文件。因此，创建一个程序文件作为该项目的主文件。在【项目管理器】窗口的【代码】选项卡中，选择【程序】项，单击【新建】按钮。打开程序设计器，设计系统程序。生成的程序文件为“主程序.prg”，代码如下：

```
SET TALK OFF
SET STAT OFF
CLEA
SET DEFA TO d:\图书管理
DO FORM 主界面
READ EVEN
```

## 8. 编译应用程序

打开项目文件“图书管理.pjx”，并在连编生成“图书管理系统.exe”文件后运行。具体编译步骤如下。

- (1) 单击【项目管理器】窗口中的【连编】按钮，弹出【连编选项】对话框。
- (2) 在对话框中选择【连编可执行文件】单选项，然后单击【确定】按钮。
- (3) 在弹出的【另存为】对话框中，选择适当的存储路径，并在【文件名】文本框中输入“图书管理系统”。
- (4) 然后单击【保存】按钮。

单击【项目设计器】窗口右上角的关闭按钮，关闭项目设计器。执行主菜单的【程序】→【运行】命令，并在弹出的【运行】对话框中双击“图书管理系统.exe”文件名，即可进入该数据库应用系统主界面。

## 9. 运行应用程序

- (1) 双击应用程序文件“主程序.prg”，系统显示主界面表单，如图 10.43 所示。



图 10.43 主界面表单的运行结果

(2) 单击“系统登录”菜单，进入密码验证界面，输入用户名和口令，如果回答正确，单击【确定】按钮后即可激活应用系统的菜单，如图 10.44 所示。



图 10.44 应用系统的菜单

(3) 选择“图书查询”菜单项，弹出下一级子菜单，选择“书名查询”，系统运行“书名查询.scx”，如图 10.45 所示。

图 10.45 运行结果

(4) 其他界面可一一显示。

## 思考与练习 10

### 一、思考题

1. 数据库和数据表有什么不同？各在系统中的作用是什么？
2. 查询和视图有什么不同？各有什么用途？
3. 说明为顶层表单添加菜单的步骤。
4. 在系统中，浏览报表和打印报表的属性设置的区别是什么？
5. 说明编译系统应用程序的步骤。

### 二、选择题

1. Visual FoxPro 的“参照完整性”中“插入规则”包括的选项是\_\_\_\_。  
A. 级联和忽略      B. 级联和删除      C. 级联和限制      D. 限制和忽略
2. 下列表达式中，返回结果为“.F.”的是\_\_\_\_。  
A. AT("A", "BCD")      B. "[信息]"\$ "管理信息系统"  
C. ISNULL(.NULL.)      D. SUBSTR("计算机技术", 3, 2)
3. 在 Visual FoxPro 中，如果在表之间的联系中设置了参照完整性规则，并在删除规则中选择“限制”，则当删除父表中的记录时，系统反应是\_\_\_\_。

- A. 不做参照完整性检查                      B. 不准删除父表中的记录  
C. 自动删除子表中所有相关的记录          D. 若子表中有相关记录, 则禁止删除父表中记录
4. 在 Visual FoxPro 中, 下列关于表的叙述正确的是\_\_\_\_\_。
- A. 在数据库表和自由表中, 都能给字段定义有效性规则和默认值  
B. 在自由表中, 能给字段定义有效性规则和默认值  
C. 在数据库表中, 能给字段定义有效性规则和默认值  
D. 在数据库表和自由表中, 都不能给字段定义有效性规则和默认值
5. 在 Visual FoxPro 中, 如果希望跳出 SCAN...ENDSCAN 循环体, 执行 ENDSCAN 后面的语句, 应使用\_\_\_\_\_。
- A. LOOP                      B. EXIT                      C. BREAK                      D. RETURN
6. 从内存中清除内存变量的命令是\_\_\_\_\_。
- A. RELEASE                      B. DELETE                      C. ERASE                      D. DESTROY
7. 操作对象只能是一个表的关系运算是\_\_\_\_\_。
- A. 连接和选择                      B. 连接和投影                      C. 选择和投影                      D. 自然连接和选择
8. 如果设定 LOCAL data, 那么 data 的初值是\_\_\_\_\_。
- A. 整数 0                      B. 不定值                      C. 逻辑真                      D. 逻辑假
9. 扩展名为.pjx 的文件是\_\_\_\_\_。
- A. 数据库表文件                      B. 表单文件                      C. 数据库文件                      D. 项目文件
10. 下列程序执行以后, 内存变量 y 的值是\_\_\_\_\_。

```
x=34567
y=0
DO WHILE x>0
    y=x%10+y*10
    x=int(x/10)
ENDDO
```

- A. 3456                      B. 34567                      C. 7654                      D. 76543

### 三、填空题

- 在奥运会游泳比赛中, 一个游泳运动员可以参加多项比赛, 一个游泳比赛项目可以有多个运动员参加, 游泳运动员与游泳比赛项目两个实体之间的关系是\_\_\_\_\_关系。
- 在 Visual FoxPro 中, 可以在表设计器中为字段设置默认值的表是\_\_\_\_\_表。
- 执行命令 A = 2005/4/2 之后, 内存变量 A 的数据类型是\_\_\_\_\_型。
- 诊断和改正程序中错误的工作通常称为\_\_\_\_\_。
- 问题处理方案的正确而完整的描述称为\_\_\_\_\_。
- 可以在项目管理器的\_\_\_\_\_选项卡下建立命令文件(程序)。
- 在 Visual FoxPro 中, 所谓自由表就是那些不属于任何\_\_\_\_\_的表。
- 不带条件的 DELETE 命令(非 SQL 命令)将删除指定表的\_\_\_\_\_记录。
- 在表单设计器中可以通过\_\_\_\_\_工具栏中的工具快速对齐表单中的控件。
- 为了在报表中插入一个文字说明, 应该插入一个\_\_\_\_\_控件。



## 参 考 文 献

- [1] 周山芙. Visual FoxPro 程序设计(二级)教程. 北京: 清华大学出版社, 2002.
- [2] 李雁翎. Visual FoxPro 应用基础与面向对象程序设计教程. 北京: 高等教育出版社(第二版), 2002.
- [3] 章立民. Visual FoxPro 6.0 程序设计与应用. 北京: 中国铁道出版社, 2003.
- [4] 教育部考试中心. 全国计算机等级考试大纲(2008 年版). 北京: 高等教育出版社, 2008.
- [5] 教育部考试中心. 教育部考试中心. 全国计算机等级考试二级教程——Visual FoxPro 程序设计. 北京: 高等教育出版社, 2008.
- [6] 李淑华. Visual FoxPro 程序设计实训与考试指导. 北京: 高等教育出版社, 2004.
- [7] 沈大林. Visual FoxPro 程序设计案例教程. 北京: 中国铁道出版社, 2004.
- [8] 相万让. Visual FoxPro 程序设计知识与练习. 北京: 人民邮电出版社, 2005.
- [9] 相万让. Visual FoxPro 程序设计实训与实例. 北京: 人民邮电出版社, 2005.
- [10] 王松, 刘强. Visual FoxPro 数据库开发经典案例解析. 北京: 清华大学出版社, 2005.
- [11] 刘丽. Visual FoxPro 程序设计. 北京: 中国铁道出版社, 2006.
- [12] 沈蒙波, 苏术锋. Visual FoxPro 程序设计. 北京: 中国计划出版社, 2007.
- [13] 史济民, 汤观全. Visual FoxPro 及其应用系统开发. 北京: 清华大学出版社, 2004.
- [14] 刘淳. Visual FoxPro 数据库与程序设计(第 2 版). 北京: 中国水利水电出版社, 2007.
- [15] 黎能武. Visual FoxPro 6.0 程序设计教程(第 2 版). 北京: 中国水利水电出版社, 2008.